

WPS-PLUS/PC

List and Sort Processing

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Copyright © 1985 Digital Equipment Corporation. All rights reserved.

The postage prepaid Reader's Comments form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

digital™

ALL-IN-1

DEC

DECmate

DECpage

DECspell

DECsystem-10

DECSYSTEM-20

DECUS

DECwriter

DIBOL

MASSBUS

PDP

P/OS

Professional

Rainbow

RSTS

RSX

UNIBUS

VAX

VMS

VT

Work Processor

WPS-PLUS

IBM™, PC-DOS™ and PC/XT™ are trademarks of the International Business Machines Corporation.

Poly-COM™, Poly-TRM™, and Poly-XFR™ are trademarks of Polygon Associates, Inc.

Contents

About This Guide

Purpose of This Guide	ix
Audience	ix
Prerequisites	ix
Associated Documents	x
Conventions	x

1 Introduction to List and Sort Processing

What is List Processing?	1-1
The List Document	1-2
The Form Document	1-4
The List Specification Document	1-5
The Output Document	1-5
How to Run List Processing	1-7
What is List Processing Math?	1-7
What is Sort Processing?	1-8

2 The List Document

Creating a List Document	2-2
Field Names	2-3
Field Values	2-3

What Fields to Include	2-4
Order of Fields	2-4
Missing Fields and Field Values	2-5
Inserting Comments	2-5
Maintaining List Documents	2-6
List Size	2-6
Order of Records	2-6
Guidelines for Editing List Documents	2-6
Using Abbreviation and Library Documents	2-7

3 The Form Document

Creating a Form Document	3-2
Types of Form Documents	3-3
Letters	3-3
Envelopes	3-7
Labels	3-9
Creating Lists, Directories, and Tables	3-10
Leaders and Trailers	3-12
Spacing Leaders and Trailers	3-14
Grouping Table Entries	3-14
Spacing Table Entries	3-16
Columns	3-17
Skipping Records	3-18
Making Sublists	3-19
Summary of Commands	3-20
Changing List Document Field Names	3-20

4 Selecting the Records to Process

Creating a List Specification Document	4-2
Restricting the Records You Process	4-2
Using OR Conditions to Test Values	4-3
Using the AND Condition to Test Values	4-4
Using BUT NOT IF to Reject Records	4-5
Using More Than One IF Statement	4-5
Testing for Empty Fields	4-6
Testing with Wild Cards	4-7
The <?> (Question Mark) Wild Card	4-7
The <*> (Asterisk) Wild Card	4-8

Testing Numeric Fields	4-9
Testing Character-for-Character	4-10
Testing for Digits Only in a Nonnumeric Field	4-10
Testing Numeric Ranges	4-11
Testing Dates	4-12
Testing Dates Against a Range	4-13
List Specification Guidelines	4-13
Testing a List Specification Document	4-14
Words and Phrases in a List Specification: Summary	4-15

5 List Processing Math

List Processing Math and Editor Math	5-1
Using List Processing Math	5-2
Processing List Processing Math	5-4
Starting List Processing Math	5-4
How to Create a FORMULA Control Block	5-5
Stopping List Processing Math	5-6
List Processing Math Commands	5-7
FORMULA Command Guidelines	5-7
Allowable Characters	5-8
Constants	5-9
List Processing Math: Complete Example	5-9
Averages	5-11
Totals and Subtotals	5-14
Truncation and Rounding	5-18
Default Format	5-19
Customized Format	5-20
Decimal Fractions	5-20
Commas	5-21
Dollar Signs	5-21
Trailing Minus Sign	5-21
Leading Plus Sign	5-22
Parentheses	5-22
Asterisks	5-22
List Processing Math Error Messages	5-23
FORMULA Command Checklist	5-24
Format Checklist	5-25
Input Field Value Checklist	5-25

6 Sorting a List Document

- Using Sort Processing 6-1
- Sort Specification Documents 6-2
 - How Sort Processing Works 6-2
 - How to Create a Sort Specification Document 6-4
 - How to Run Sort Processing 6-4
- Sorting on More Than One Field 6-5
- Examples of Sort Output Documents 6-7
- Sort Alphabet 6-14
- Sorting Missing Fields or Field Values 6-15
- Words and Phrases in a Sort Specification: Summary 6-15

7 Running List and Sort Processing

- How to Run List Processing 7-2
 - Identifying Documents 7-4
 - Changing Start and Finish Settings 7-6
 - Selecting an Option for Your Output 7-7
 - Validating Your Selections 7-8
 - Information Displayed During List Processing 7-10
- How to Test a List Specification Document 7-10
- How to Run Sort Processing 7-11
- How to Test a Sort Specification Document 7-12
- How to Print an Output Document 7-13

Figures

- 1-1 How List Processing Works 1-6
- 3-1 A Form Document 3-2
- 3-2 Leaders and Trailers 3-12
- 3-3 Changing the Spacing of Leaders and Trailers 3-14
- 4-1 Words and Phrases in a List Specification 4-15
- 5-1 Sample FORMULA Command Control Block 5-3
- 5-2 List Processing Math: Complete Example 5-10
- 5-3 Sample List Document Showing Subtotals 5-16
- 6-1 How Sort Processing Works 6-3
- 6-2 List Sorted on <STATE>, <CITY>, and <STREET> in Ascending Order 6-6
- 6-3 List Sorted on <STATE> and <LAST> in Ascending Order 6-8

6-4	List Sorted on <ITEM#> in Ascending Order	6-10
6-5	List Sorted on <PAYMT> Descending and <LAST> Ascending	6-11
6-6	List Sorted on <:UNIT PRICE> in Descending Order	6-13
6-7	Words and Phrases in a Sort Specification	6-15
7-1	List Processing Menu	7-2
7-2	List Processing Input Documents Form	7-5
7-3	List Processing Job Description Form	7-8

Tables

3-1	Single and Double Spacing with <!S> and <!E>	3-17
3-2	Form Document Commands	3-20
4-1	Words and Phrases in a List Specification	4-16
5-1	Editor Math and List Processing Math	5-2
5-2	List Processing Math Commands	5-7
5-3	Allowable Characters	5-8
6-1	Sort Alphabet	6-14
6-2	Words and Phrases in a Sort Specification	6-16
7-1	List Processing Output Document Menu Options	7-7

About This Guide

Product Name: WPS-PLUS/PC Version 1.0

Purpose of This Guide

WPS-PLUS/PC List and Sort Processing explains how to use WPS-PLUS List Processing, List Processing Math, and Sort Processing.

List Processing, List Processing Math, and Sort Processing are features of WPS-PLUS. For information about the WPS-PLUS Editor, see *WPS-PLUS Editor Functions*.

Audience

This guide is for users who want to produce documents such as form letters, mailing lists, and monthly financial reports.

Prerequisites

Before using List and Sort Processing, you should complete the exercises in *WPS-PLUS Getting Started*. No previous List Processing experience is necessary.

Associated Documents

For details about WPS-PLUS you can also refer to the following documents:

WPS-PLUS/PC Installation and Printer Tables

WPS-PLUS/PC Getting Started

WPS-PLUS on PC

WPS-PLUS/PC Editor Functions

WPS-PLUS/PC Quick Lookup

WPS-PLUS Glossary

Conventions

ALL CAPS	Identifies a WPS-PLUS editor function. For example: Press ADVANCE PARA means to press the ADVANCE key and then the PARA key.
ALT/x	Means hold down the ALT key while you press a function key. For example: Press ALT/INTERRUPT.
CTRL/x	Means hold down the CTRL key while you press the designated letter key. For example: Press CTRL/W.
Dot Matrix	Indicates text that appears on your screen.
Enter	Means type your response and press a terminator key, such as RETURN or TAB.
<field name>	Indicates a field name for List Processing or Sort Processing.
Gold x	Means press the Gold key, labeled Num Lock on some keypads, and then a function key. For example: Press Gold FILE DOCMT.
(RETURN)	Indicates that you press the RETURN key, in an example.

SHIFT/x	Means hold down the SHIFT key while you press a function key. For example: Press SHIFT/DO.
WPS-PLUS	Means WPS-PLUS/PC.

Introduction to List and Sort Processing

This chapter introduces:

- List Processing
- Sort Processing
- List Processing Math

What is List Processing?

Suppose you need to send a form letter to forty people. Each letter is identical except for the recipient's name in the salutation and the address. Rather than typing the letter forty times, you can use List Processing to merge a list of the names and addresses with the form letter. The result is a personalized letter for each person on the list.

You can use List Processing to:

- Create personalized letters or reports describing a new product or service
- Reformat items in a list so they appear in a table, letter, report, or directory
- Perform calculations and insert the results in reports
- Create sublists from master lists without retyping the lists
- Track the sales of all or selected products in your inventory

Before you use List Processing, you must create three documents:

- A list document
- A form document
- A list specification document

You create these documents the same way you create other documents with WPS-PLUS. When you run List Processing, WPS-PLUS uses these documents to produce the output document.

The List Document

The list document contains a list of records. For example, if you are producing individualized letters, your list document contains a list of the names and addresses of the people who will receive the letter. Each name and address constitutes a record in the list.

This is how the records are arranged in the list document:

```
<TITLE>Mr.  
<NAME>Benson  
<FIRST>Robert  
<STREET>156 High Pine Ave  
<CITY>Boston  
<STATE>MA  
<ZIP>02739  
<>
```

```
<TITLE>Ms.  
<NAME>Lewis  
<FIRST>Beth  
<STREET>14 Abolition St  
<CITY>Atkinson  
<STATE>NH  
<ZIP>03033  
<>
```

```
<TITLE>Mr.  
<NAME>Hayes  
<FIRST>Peter  
<STREET>934 12th Street  
Apt 2C  
<CITY>Philadelphia  
<STATE>PA  
<ZIP>19103  
<>
```

The information in angle brackets is the field name. The field name is followed by the field value. Together, the field name and field value are the field. Note that a field can be more than one line long. The entire block of fields for each person is called a record. The two angle brackets, (<>), mark the end of the record.

field name field value

↓ ↙

```
<TITLE>Ms.  
<NAME>Lewis  
<FIRST>Beth  
<STREET>14 Abolition St  
<CITY>Atkinson  
<STATE>NH  
<ZIP>03033  
<>
```

} record

MK-02287-00

Chapter 2 describes how to create and use list documents.

The Form Document

The form document is the text, such as a letter or report, you want to personalize. A form document includes field names that match the field names you want to include from the list document. For example:

```
----- NEW PAGE -----  
Dear <TITLE> <NAME>,  
  
The East Coast Bank is Pleased to announce a new checking Plan that is  
available to all <CITY> residents. We would like to take this  
oPPortunity...
```

When you run List Processing, the field values *Mr.*, *Benson*, and *Boston*, from the list document are inserted where the *<TITLE>*, *<NAME>*, and *<CITY>* field names appear in the form document. See Figure 1-1.

Chapter 3 describes how to create and use form documents.

The List Specification Document

The list specification document tells List Processing what records to use from the list document. The list specification document can specify that all records are to be processed, or only those records with certain field values. Even if you want all records processed, you must create this document.

You can process all records in your list document with the following list specification document:

```
PROCESS RECORD
```

You can put statements in the list specification document that restrict the records List Processing uses. For example, you can send letters to only those people who live in Boston. The list specification document you create would look like this:

```
IF <CITY>=BOSTON  
THEN PROCESS RECORD
```

Chapter 4 describes all the statements you can use in a list specification document.

The Output Document

List Processing uses your list document, form document, and list specification document to create an output document. For example, if you run List Processing to create forty personalized letters, the output document contains forty letters. There is one letter for each record in the list document. If you read the output document, you see a NEW PAGE mark between each letter. When you print the output document, each letter is printed on a separate page.

The output document in Figure 1-1 is created when List Processing inserts a title, name, and city from a list document record into each copy of the form letter. The list specification document restricts the processing to those records with Boston in the <CITY> field. Thus, List Processing creates form letters for Boston residents only.

The list document:

```
<TITLE> Mr.  
<NAME> Benson  
<FIRST> Robert  
<STREET> 156 High Pine Ave  
<CITY> Boston  
<STATE> MA  
<ZIP> 02739  
<>
```

```
<TITLE> Ms.  
<NAME> Lewis  
<FIRST> Beth  
<STREET> 14 Abolition St  
<CITY> Atkinson  
<STATE> NH  
<ZIP> 03033  
<>
```

```
<TITLE> Mr.  
<NAME> Hayes
```

The form document:

```
Dear <TITLE> <NAME>,  
The East Coast Bank is pleased to announce a new checking  
plan that is available to all <CITY> residents. we would  
like to take this opportunity...  
----- NEW PAGE -----
```

The list specification document:

```
IF <CITY>=BOSTON  
THEN PROCESS RECORD
```

The output document:

```
Dear Mr. Benson,  
The East Coast Bank is pleased to announce a new checking  
plan that is available to all Boston residents. we would  
like to take this opportunity...
```

MK-01661-00

Figure 1-1 How List Processing Works

How to Run List Processing

After you create the form, list, and list specification documents, you can run List Processing. The following is an overview of the steps you follow. More information is provided in Chapter 7.

To run List Processing:

- 1 Enter LP (List Processing) from the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).
The List Processing Menu appears.
- 2 Enter DOC (Document Processing).
The Input Documents form displays.
- 3 Follow the instructions at the bottom of your screen to run List Processing.

You can use the P (Print) option from the Document Processing Menu to print the output document.

What is List Processing Math?

List Processing Math allows you to perform calculations on numerical information in your list document. List Processing Math can add, subtract, multiply, and divide.

You can use List Processing Math to:

- Produce weekly business reports for a company
- Select customers who are behind on payments, and produce letters requesting payment
- Produce annual sales reports
- Produce weekly payroll reports

To use List Processing Math, you create:

- A list document

The list document contains records with some numerical fields, and control blocks containing mathematical formulas to be performed on these fields.

- A form document
- A list specification document

For information about List Processing Math, see Chapter 5.

What is Sort Processing?

Sort Processing arranges lists of alphabetic or numeric information in ascending or descending order. You use the sorted list as input for List Processing.

You can use Sort Processing to:

- Sort a mailing list by zip codes
- Alphabetize a file by sorting on customers' last names
- Maintain a filing system

Hint. You can add new entries to the beginning or the end of a list, then run Sort Processing to place the new entries in sequence with the old ones.

To use Sort Processing, you create:

- A list document

The list document contains the records to be sorted. You can sort a list document you have created for List Processing.

- A sort specification document

The sort specification document tells Sort Processing which record field to use in sorting records, and which order to sort in (ascending alphabetic/numeric order or descending alphabetic/numeric order).

When you run Sort Processing, you produce a sort output document. The sort output document is a sorted list.

For information about Sort Processing, see Chapter 6.

2

The List Document

The list document is a series of records. List Processing takes information from the list document records and the form document and produces the output document. The output document contains a separate copy of the form document for each list document record. This chapter explains how to create and use list documents. Helpful guidelines include:

- List document size
- Order of records in lists
- Order of fields in records
- Using abbreviation and library documents
- The effects of missing fields and field values

Creating a List Document

To create a List document:

- 1 Enter C (Create) from the Document Processing Menu and fill out the Create form.

When you press RETURN, the screen clears and you are in the editor.

- 2 Type a left angle bracket (<), the name for the field, and a right angle bracket (>). The right angle bracket completes the first field name of the first record.
- 3 Type the value for that field immediately after the right angle bracket. For example:

```
<NAME>Marshall's Chimney Service
```

Note that the field name and field value combined are one field.

- 4 Repeat steps 2 and 3 until you have typed all field names and values for the record.
- 5 End the record with an end record marker, by typing left and right angle brackets (<>). For example:

```
<NAME>Marshall's Chimney Service
<ADDRESS>1492 Brookdale Dr.
Corvallis, OR 75603
<LAST ORDER>50 gallons creosote solvent
<>
```

- 6 Press RETURN twice to put a blank line between each record.
The blank line between records is not necessary, but it makes your list document easier to read.
- 7 Repeat steps 2 through 6 until you have typed all records in the list document.
- 8 Press Gold FILE DOCMT to file the list document.

Field Names

Use these guidelines when you create field names:

- A field name starts with a left angle bracket (<) and ends with a right angle bracket (>).
- A field name cannot contain additional angle brackets.
- A field name cannot contain an exclamation point (!). An exclamation point inside angle brackets indicates a command. This topic is discussed in the Leaders and Trailers section in Chapter 3.
- A field name can contain uppercase or lowercase letters. The field names <ADDRESS>, <Address>, and <address> are the same.
- Field names in the list, form, and list specification documents must match. For example, if you use <ADDRESS> in your list document, you cannot use <ADDR> in your form document or your list specification document to refer to the same field.

Field Values

Use these guidelines when typing field values to ensure that the information will be correct in your output document:

- Make sure you type field values in a list document exactly as you want them to appear in your output document. For example, do not use all uppercase letters unless you want uppercase letters in the output document.
- You can type several lines of information after one field name. For example:

```
<ADDRESS>12 Hill Street  
Grantham,  
Lincolnshire,  
GR1 0bt  
England
```

You can put RETURNS in field values. In the previous example, *12 Hill Street* is separated from *Grantham* by a RETURN.

When List Processing inserts the value of <ADDRESS> into the output document, it inserts a RETURN after *Street* so *Grantham* is on a separate line.

- Do not type a space or a tab before a field value unless you want the extra spacing. If you type a tab in front of a field value, your output document will also have the tab spacing. For example, if you put a tab in front of *Grantham* in the last example, the output document looks like this:

```
12 Hill Street
      Grantham,
Lincolnshire,
GR1 0bt
England
```

What Fields to Include

The information you want to insert in the form document determines what fields you need in the list document. To make your list document versatile, include all the fields you anticipate needing for present and future form documents.

You can include more fields in a list document than you need in any one form document. List Processing uses only the fields specified in the form document.

Order of Fields

You can arrange fields in a list document record in any order. However, lists are easier to maintain if you enter fields in the same order for each record.

Missing Fields and Field Values

Be careful not to omit any part of a field from a record. If a field value or a field name is missing from a record, List Processing cannot insert the value in the output document. List Processing does, however, insert a blank where the value would go.

Inserting Comments

You can put comments in a list document. Comments help others who use your list by explaining what your list contains or how to process the list.

Here is an example of a useful comment:

```
----- START CONTROL -----  
COMMENT  
  
This is the customer mailing list, CUSTN1, for customers A-M.  
Each record has the fields <TITLE>, <FIRST>, and <LAST> for a  
customer's name, and <ADDRESS> for a customer's complete mailing  
address. The <DOWED> field contains the amount a customer owes  
at the end of each month. See document CUSTN2 for customers N-Z.  
----- END CONTROL -----
```

Use a control block to include a comment if:

- The comment must be easy to find
- The comment contains a field name
- The comment is between the first field in a record and the end record marker

To put a comment in a control block:

- 1 Press Gold CMND to start the control block.
- 2 Type COMMENT and press RETURN.
- 3 Type the comment.
- 4 Press Gold CMND to end the control block.

You can type comments without using a control block. Be sure the comment is after the end record mark (<>) and before the first field of the next record. Remember, if you need to include field names in a comment, put the comment in a control block.

Maintaining List Documents

The following are suggestions to help you maintain your list documents.

List Size

You can divide a very long list document, for example one with 500 records, into shorter ones that are easier to use. Instead of creating one list document for all customers, create one list document for customers whose last names begin with letters A through D, another document for customers E through H, and so on.

Order of Records

If the order of the records in the list document is important, insert each new record in the correct location. You can also use Sort Processing to arrange your list. See Chapter 6. List Processing processes records in the order they appear in the list document.

Guidelines for Editing List Documents

- If there is a blank line between records, you can use the PARA key to move from one record to the next when editing the list document.
- To avoid retyping the field names, create a blank record with just the field names. Use Gold CUT to put the blank record in the paste area. Then press SHIFT/PASTE to make copies of the record. Go back and fill in field values.
- Use CUT to remove records from the list. Use CUT and SHIFT/PASTE to move records from one place to another.

- Use the ENTER key to advance or back up to the start of a field value.

NOTE: You can include multinational and technical characters in any document used in List or Sort Processing. However, your printer must have the capability to print those characters. See your Software Product Description for more information about the hardware requirements for printing these characters.

Using Abbreviation and Library Documents

If you use List Processing often, store the field names for future records in an abbreviation or library document. For information about abbreviation and library documents, see *WPS-PLUS Editor Functions*.

3

The Form Document

A form document is text, such as a letter or report, that includes field names that match those in the list document. When you run List Processing to create the output document, the field names are replaced with the field values from the list document. This chapter explains how to:

- Create form documents to produce different types of output
- Put leaders and trailers in an output document
- Single-space, double-space, and group table entries in an output document
- Skip processing list document records with a form document command
- Create sublists with a form document command
- Change field names in list document records

Figure 3-1 shows an example of a form document:

April 1, 1986

<TITLE> <FIRST> <LAST>
<ADDRESS>

Dear <TITLE> <LAST>:

According to our records, <TITLE> <LAST>,
you have flown <MILES> miles with us. As
one of our most valued customers, we would
like to tell you about special discounts
available to you starting September 15 . . .

MK-02285-00

Figure 3-1 A Form Document

Creating a Form Document

To create a form document:

- 1 Enter C (Create) from the Document Processing Menu and fill out the Create form.

When you press RETURN, the screen clears and you are in the editor.

- 2 Press Gold NEW PAGE to start the form document.

The NEW PAGE mark appears in the output document between copies of the form document. When you print the output document, each copy of the form document begins on a new page.

If the form document is the template for entries in a table, do not insert a NEW PAGE mark.

- 3 Type the form document. Put field names where you want items from the list document to be inserted. Type each field name exactly as it appears in the list document.

You can change the ruler, include bolding and underlining, and use other WPS-PLUS functions.

- 4 Press Gold FILE DOCMT to file the form document.

Types of Form Documents

The following sections explain how to create form documents for:

- Letters
- Envelopes
- Labels
- Tables

Letters

When you create a form letter, you include field names that match the field names in the list document. You do not have to insert blank lines at the top of the form letter to allow for a letterhead. When you are ready to print the letter, use print settings to position the text below the letterhead.

To create a form letter for letterhead stationery:

- 1 Create a new document.
- 2 Press Gold NEW PAGE to insert a NEW PAGE mark.
- 3 Create a ruler that contains a left margin (L) where you want the salutation to start. Put a tab (T) in the ruler where you want the date and closing of the letter to start.

NOTE: An alternative to controlling the left margin with a ruler is to use print settings. See Chapter 4 in WPS-PLUS on PC for information about print settings.

- 4 Type the date directly under the ruler at the tab position.

- 5 Type the field names for the recipient's address.

Include punctuation and spacing when typing the field names. See the following example.

- 6 Type the salutation, followed by the field names you want.
- 7 Type the text of the form letter, including the desired field names.

Each field name in the form document must match a field name in the list document. However, List Processing allows you to enter field names in uppercase or lowercase letters. For example, you can have <STATE> in the list document and <state> in the form document.

- 8 Type the closing.
- 9 Press Gold FILE DOCMT to file your document.

In this letter, note that the field names are separated by spaces and necessary punctuation is included:

```
L-----T-----R
----- NEW PAGE -----
                        April 1, 1986
```

```
<TITLE> <FIRST> <LAST>
<STREET>
<CITY>, <STATE> <ZIP>
```

Dear <TITLE> <LAST>:

All of us at the Southport Bookstore would like to thank you,
<TITLE> <LAST>, for your patronage this year.

Sincerely,

William Howard

When you run List Processing, WPS-PLUS repeats the letter for each record in the list document. Each time the letter is produced, a different title and last name are inserted into the letter.

Here are sample list document records:

<TITLE>Mrs.,
<FIRST>Deborah
<LAST>Howe
<STREET>444 Evans Road
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>

<TITLE>Mrs.,
<FIRST>Beth
<LAST>Morris
<STREET>737 Poplar Avenue
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>

After processing, your output document contains these letters:

----- NEW PAGE -----
April 1, 1986

Mrs. Deborah Howe
444 Evans Road
Londonderry, NH 03053

Dear Mrs. Howe:

All of us at the Southport Bookstore would like to thank you,
Mrs. Howe, for your patronage this year.

Sincerely,

William Howard

(continued)

----- NEW PAGE -----

April 1, 1986

Mrs. Beth Morris
737 Poplar Avenue
Londonderry, NH 03053

Dear Mrs. Morris:

All of us at the Southport Bookstore would like to thank you,
Mrs. Morris, for your patronage this year.

Sincerely,

William Howard

If your form document has more than one page, use any of the methods discussed in *WPS-PLUS Editor Functions* to paginate your form document. You can edit your output document to remove the first NEW PAGE mark.

To print the letter with the text positioned under the letterhead:

- 1 Enter P (Print) from the Document Processing Menu.
- 2 Follow the instructions to change print settings. Change the lines in the top margin to the number of lines you want between the top of the page and the date. You can determine this number by measuring the distance from the top of the page to where you want the date to appear on a blank sheet of paper. (If you are using a letter quality printer, there are six lines per inch.)

In this example, the lines in the top margin were set to 12 to allow for the letterhead.

- 3 Press RETURN to print the output document.

When you print the output document each letter prints on a separate page.
The first letter looks like this:



SouthPort Bookstore
100 Beacon St.
SouthPort, NH 03039

April 1, 1986

Mrs. Deborah Howe
444 Evans Road
Londonderry, NH 03053

Dear Mrs. Howe:

All of us at the SouthPort Bookstore would like to thank you,
Mrs. Howe, for your patronage this year.

Sincerely,

William Howard

MK-02289-00

Envelopes

When you use List Processing to address envelopes, use continuous-form envelopes in your printer. Continuous-form envelopes are either attached to continuous-form paper or are attached end-to-end.

To create a form document that prints the return address and the recipient's address on the envelope:

- 1 Create a new document.
- 2 Press Gold NEW PAGE to put a NEW PAGE mark at the top of the form document.

If you do not put a NEW PAGE mark at the beginning of the form document, the first envelope will have the return address printed one line higher than all subsequent envelopes.

- 3 Create a ruler with the left margin (L) where you want the return address and a tab (T) where you want the address from the list document to start.

NOTE: An alternative to controlling the left margin with a ruler is to use print settings. You can also adjust the top margin with a print setting. See Chapter 4 in WPS-PLUS on PC for information about print settings.

- 4 Type the return address directly under the ruler.
- 5 Press RETURN until the cursor is at the beginning of the line for the recipient's address.
- 6 Tab to where the address should go.
- 7 Type the field names for name and address.

Remember to put a space between field names on the same line.

- 8 Press Gold FILE DOCMT to file your document.

For example:

```
L-----T-----R
----- NEW PAGE -----
SouthPort Bookstore
100 Beacon St.
SouthPort, NH    03039
```

```
<TITLE> <FIRST> <LAST>
<STREET>
<CITY>, <STATE>    <ZIP>
```

Your output document contains this information after processing:

----- NEW PAGE -----

Southport Bookstore
100 Beacon St.
Southport, NH 03039

Mrs. Deborah Howe
444 Evans Road
Londonderry, NH 03053

----- NEW PAGE -----

Southport Bookstore
100 Beacon St.
Southport, NH 03039

Mrs. Beth Morris
737 Poplar Avenue
Londonderry, NH 03053

Labels

Labels, like envelopes, must be attached to continuous-form paper to print them using List Processing.

To format a form document for continuous-form labels:

- 1 Create a ruler in the form document with left and right margins.
The number of spaces between the L and R margin settings must be less than or equal to the number of spaces that fit on your label.
- 2 Press Gold NEW PAGE to insert a NEW PAGE mark.
- 3 Move the cursor to where the text on the label should start.
- 4 Type the field names for title, first name, last name, and address.
Remember to put a space between field names on the same line.
- 5 Press Gold FILE DOCMT to file your document.

For example, this is the form document for a label:

```
L-----R-----  
----- NEW PAGE -----  
<TITLE> <FIRST> <LAST>  
<STREET>  
<CITY>, <STATE>   <ZIP>
```

Your output document looks like this after processing:

```
----- NEW PAGE -----  
Mrs. Deborah Howe  
444 Evans Road  
Londonderry, NH   03053  
----- NEW PAGE -----  
Mrs. Beth Morris  
737 Poelar Avenue  
Londonderry, NH   03053
```

Creating Lists, Directories, and Tables

If you create a mailing list, directory or table, you probably do not want a new page for each record. For example, a directory has the name, address and telephone information repeated continuously down the pages. Each record for the directory is printed one after another until the page is filled. Then List Processing goes on to the next page and continues printing until the last record is processed.

To print a list or table:

Omit NEW PAGE marks in the form document.

For example, you can create a mailing list from a list document of customer records. The form document contains only the field names that appear in the mailing list. Suppose each list document record contains the following fields:

```
<FIRST>
<MIDDLE>
<LAST>
<STREET>
<CITY>
<STATE>
<ZIP>
<TELEPHONE>
<CREDIT>
<LAST UPDATED>
```

Your form document contains the fields that appear in mailing list entries. Note that there is no NEW PAGE mark in the form document.

```
<LAST>, <FIRST>
<STREET>
<CITY>, <STATE> <ZIP>
```

Be sure to press RETURN twice after you type each record in your form document. This appears as a blank line in the output document between each of the records. Your output document is a mailing list in which a blank line separates each of the address records.

For example:

```
Collins, Nancy
649 Hawthorne Road
Bloomington, IN    47041
```

```
Dumas, Andre
19 Walnut Street
Bloomington, IN    47041
```

```
Roberts, William
30 First Street
Bloomington, IN    47041
```

Leaders and Trailers

When creating a table or directory, you may want to label the columns. Items such as titles and column headings that precede repeated field names are called leaders. Information that follows repeated field names is a trailer.

For example, Figure 3-2 is a parts list with a title and column labels for a leader, and ordering information for a trailer.

PARTS LIST			}	Leader
<u>Part</u> <u>Number</u>	<u>Part Name</u>	<u>Cost</u>		
80-0030	2" brush	1.15		
80-0040	drop cloth	0.69		
80-0050	latex gloves	0.12		
80-0235	plaster	0.79		
80-0082	roller	0.89		
Parts can be ordered from Stockroom 5.			}	Trailer

MK-02286-00

Figure 3-2 Leaders and Trailers

To set off the leaders and trailers from the area to be repeated, use the `<!S>` (start repetition command) before the area to be repeated. The `<!S>` command signals the start of field names and text to be processed repeatedly. The portion of the form document to be repeated until all records are processed is called the repetitive area. If you have a trailer, type the `<!E>` (end repetition command), following the area to be repeated.

The form document for the parts list in Figure 3-2 looks like this:

```

L-----T-----T-----R
                                PARTS LIST

Part
Number          Part Name          Cost
-----
L-----T-----,-----R
<!S>
<PART#>          <NAME>   <:COST><!E>

Parts can be ordered from Stockroom 5.
```

NOTE: The second ruler in the form document contains a decimal-aligned tab (.) under the Cost column. Using this tab aligns all of the decimal points in the values appearing in this column in your output document. For information about decimal aligned tabs, see Chapter 3 of *WPS-PLUS Editor Functions*.

When processing starts, the leader is written to the output document. Then processing of the repetitive area of the form document begins. After processing all records in the list document, List Processing copies the trailer into the output document and processing is completed.

To put a leader in your output document:

- 1 Type the text for the leader at the top of your form document.
 - 2 Type the <!S> command on a separate line after the leader.
- The <!S> command starts the repetitive area.

To put a trailer in your output document:

- 1 Type the <!E> command after the repetitive area in your form document.
- The <!E> command ends the repetitive area.
- 2 Type the text for the trailer after the <!E> command.

Spacing Leaders and Trailers

If the number of blank lines after the leader or before the trailer is important, check your output document before printing.

To change the spacing around leaders and trailers, add or remove blank lines in these areas:

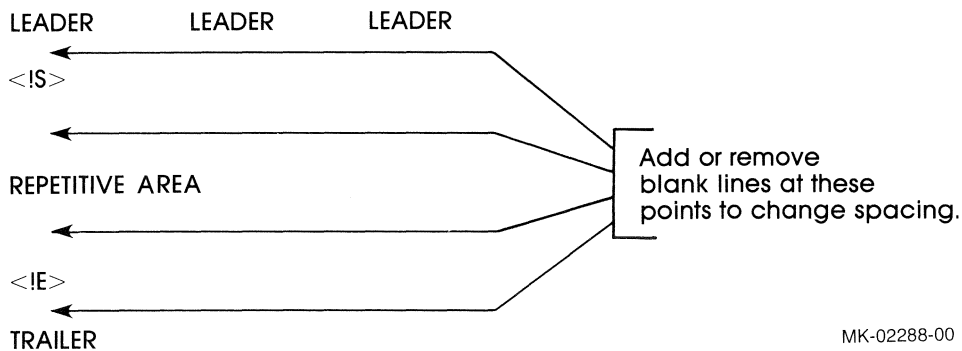


Figure 3-3 Changing the Spacing of Leaders and Trailers

Grouping Table Entries

Grouping table entries is a way to make the table more readable. To group records, you tell List Processing to put a blank line after a certain number of records.

Here is a sample output document with a blank line after every three records:

```

                                PARTS LIST

Part
Number          Part Name          Cost
-----
80-0030          2" brush            1.15
80-0040          drop cloth           0.69
80-0050          latex gloves         0.12

80-0235          plaster             0.79
80-0082          roller             0.89

```

Parts can be ordered from Stockroom 5.

Here is the form document for placing a blank line after every three items:

```

L-----T-----T-----R
                                PARTS LIST

Part
Number          Part Name          Cost
-----
L-----T-----T-----R
<!S>
<PART#>          <NAME>          <:COST><>
<PART#>          <NAME>          <:COST><>
<PART#>          <NAME>          <:COST>
<!E>

```

Parts can be ordered from Stockroom 5.

A get-next-record (<>) command begins processing of the next record. Thus, three records are processed each time the repetitive area is processed. The placement of the <!S> and <!E> commands puts a blank line after every three lines.

NOTE: A *get-next-record* command is *NOT* used after the last field because the `<!E>` command also causes *List Processing* to advance to the next record. If there was a `<>` after the last `<:COST>` field in the previous example, *List Processing* would skip every fourth record.

Spacing Table Entries

To single-space table entries if your form document does not contain an `<!S>` or an `<!E>` command, end your form document without pressing `RETURN`. For example:

```
L-----T-----R
<PART#>      <NAME>          <:COST> (Gold FILE DOCMT)
```

Each time *List Processing* begins processing the repetitive area, a table entry is started on a new line, so table entries are single-spaced.

To double-space entries in a form document that does not contain an `<!S>` or an `<!E>` command, press `RETURN` at the end of your form document.

For example:

```
L-----T-----R
<PART#>      <NAME>          <:COST>(RETURN)
```

Here is sample output:

80-0030	2" brush	1.15
80-0040	drop cloth	0.69
80-0050	latex gloves	0.12

Table 3-1 explains how to single-space or double-space table entries when you are using the `<!S>` or `<!E>` command or both.

Table 3-1 Single and Double Spacing with <!S> and <!E>

If the Form Document Contains	To Single-space Entries	To Double-space Entries
<!S> only	Put <!S> immediately before the first record field. End the repetitive area by pressing Gold FILE DOCMT.	Put <!S> on a separate line. End the repetitive area by pressing Gold FILE DOCMT.
<!E> only	Put <!E> on a separate line.	Put a blank line before <!E>.
<!S> and <!E>	Put <!S> on a separate line. Put <!E> immediately after the last record field.	Put <!S> on a separate line. Put <!E> on a separate line.

Columns

To create an output document with columns:

- 1 Put tab positions in your form document ruler.
- 2 Use the <> command between tab positions to get the next record.

Here is a form document:

```

L-----T-----T-----T-----R
PART NAME      COST      PART NAME      COST
<!S>
L-----T-----T-----R
<NAME>      <:COST><>      <NAME>      <:COST>

```

List Processing lists the records row by row. This is how the printed table looks:

PART NAME	COST	PART NAME	COST
2" brush	1.15	drop cloth	0.69
latex gloves	0.12	plaster	0.79
roller	0.89		

When you combine the parts list document and the form document, List Processing:

- Processes the first record and puts values at the left margin and the first decimal tab position
- Processes the <> command and gets the next record
- Processes the record and puts information at the other tab positions
- Returns to the left margin, gets the next record and begins the next line of the output document

Skipping Records

To process every other list document record, put a <> command in the form document before the <!E> command. If you do not have an <!E>, put the <> after the last field name.

For example:

```
L-----T-----T-----R
                        PARTS LIST

Part
Number          Part Name          Cost
-----
<!S>
L-----T-----,-----R
<PART#>          <NAME> <:COST><><!E>
```

Parts can be ordered from Stockroom 5.

The <> command sends processing to the next record. Before that record can be processed, the <!E> command sends List Processing to the next record. Consequently, every other record is skipped.

Making Sublists

Assume you have a list document with records for states in New England. If you want separate list documents for each state:

- 1 Create a form document containing the <!R> (copy record) command.
- 2 Create a list specification document that limits the records in the sublist to only one state.

See Chapter 4 for information on creating list specification documents.

You can include comments for the new list document as a leader or trailer in the form document. Here is a sample form document that produces a list document as output. The <!R> command inserts an exact copy of each selected record in the output list document:

```
This file contains active <STATE> customer accounts.
```

```
<!S><!R><!E>
```

```
This list is current as of April 1, 1986.
```

If you want blank lines between records, press RETURN after the <!R> command and put <!E> on the next line.

This list specification document restricts the records in the sublist to those with the field value Maine:

```
IF <STATE>=Maine  
THEN PROCESS RECORD
```

Note that you can also use Gold GET DOCMT to make one copy of a list document or to insert it into another document.

Summary of Commands

Form document commands change the formatting of your output document. Table 3-2 summarizes the commands you can use in a form document.

Table 3-2 Form Document Commands

Use	To
<!S>	Include leader text to be printed at the top of your output document.
<!E>	Include trailer text to be printed at the bottom of your output document.
<>	Advance to the next list document record.
<!R>	Copy records to the output document.

Changing List Document Field Names

You can change the field names in all records in a list document by running List Processing to create a new list document. You can use this procedure to remove certain fields from all list document records or to add fields to all list documents.

To change field names:

- 1 Create a form document containing the new field names.
- 2 Type two left angle brackets, <<, before the new field name.
- 3 End the new field name with a single right bracket, >.
- 4 Type the old field name after the new field name.
- 5 Run List Processing to create a new list document as your output document.

Here is a sample record from a list document:

```
<NAME>Arthur Mitchell  
<POSITION>Sr. Sales Rep  
<LOCATION>Eastern Region Sales Office  
<MAIL ADDRESS>110 Salmon Brook Rd.  
Troy, NY 22381  
<TELEPHONE>(518) 838-4853  
<>
```

Suppose you want the list document to have fewer fields and shorter field names. It may take a long time to retype the list document. In addition, retyping a document can introduce errors.

To change the field names in the list document shown above, create a form document such as:

```
<<NA><NAME>  
<<LOC><LOCATION>  
<<PHONE><TELEPHONE>  
<<>
```

Here is a sample record from the output document.

```
<NA>Arthur Mitchell  
<LOC>Eastern Region Sales Office  
<PHONE>(518) 838-4853  
<>
```


4

Selecting the Records to Process

You can process all or a selected portion of the records in a list document. A list specification document tells List Processing which records to process. For example, if you are processing a mailing list to produce address labels, you can process records only for people who live in certain cities or states. This chapter explains how to create list specification documents to:

- Process only the list document records that contain a certain value or values
- Process only the records that contain one of several values
- Reject records that contain a certain value or values
- Process only the records that do not contain certain field values
- Test the numeric value, characters, or digits in a field

Creating a List Specification Document

To create a list specification document:

- 1 Enter C (Create) from the Document Processing Menu and fill out the Create form.

When you press RETURN, the screen clears and you are in the editor.

- 2 Type PROCESS RECORD.

This list specification processes all the records in your list document.

To select only certain records for processing, use an IF statement. The IF statement and its conditions are explained in this chapter.

- 3 Press Gold FILE DOCMT to file the list specification document.

Restricting the Records You Process

To restrict the records you process, insert an IF statement in the list specification document.

IF statements have this general form:

```
IF <FIELD NAME>=FIELD VALUE  
THEN PROCESS RECORD
```

For example, the following list specification document processes records only if the <STATE> field has the value NH.

```
IF <STATE>=NH  
THEN PROCESS RECORD
```

List Processing checks each list document record to see if the <STATE> field has the value NH. Records with other values are skipped.

You can further restrict the selection by adding one or more of the following conditions to the IF statement:

- OR conditions
- AND conditions
- BUT NOT IF conditions

Using OR Conditions to Test Values

To examine a field for any of two or more values, use one or more OR conditions.

An OR condition has the form:

OR=FIELD VALUE

For example:

```
IF <STATE>=New Hampshire
    OR=NH
    OR=N.H.
THEN PROCESS RECORD
```

This example selects records referring to New Hampshire, accepting any of three values for the field <STATE>.

List Processing does NOT process a record with NEW HAMPSHIRE as the field value because the letters are all uppercase. Only field values with the same capitalization as those in the list specification document are processed.

You can also use OR conditions to accept other values:

```
IF <STATE>=New Hampshire
    OR=New Jersey
    OR=New York
THEN PROCESS RECORD
```

Do not put spaces before or after the equals sign. If you put a space after the equals sign, List Processing looks for a field value that also has a space before the first character. You do not have to indent the OR condition, but doing so improves the readability of the list specification document.

You can use OR to modify AND, BUT NOT IF, and IF. See the next sections.

Using the AND Condition to Test Values

Use the AND condition when you want to process only those records with two or more desired field values.

An AND condition has the form:

```
IF <FIELD NAME 1>=FIELD VALUE 1
    AND <FIELD NAME 2>=FIELD VALUE 2
THEN PROCESS RECORD
```

In the next example, the list specification document uses AND to find records for residents of Princeton, New Jersey:

```
IF <CITY>=Princeton
    AND <STATE>=New Jersey
THEN PROCESS RECORD
```

List Processing selects a record only if field values in the record match the values in the IF and the AND conditions.

You can use OR to modify AND. For example:

```
IF <CITY>=Princeton
    AND <STATE>=New Jersey
    OR=NJ
THEN PROCESS RECORD
```

List Processing selects records for residents of Princeton, New Jersey and accepts two values for New Jersey.

Using BUT NOT IF to Reject Records

To reject some records among those you are selecting, use a BUT NOT IF condition.

A BUT NOT IF condition has the form:

```
IF <FIELD NAME 1>=FIELD VALUE 1  
BUT NOT IF <FIELD NAME 2>=FIELD VALUE 2  
THEN PROCESS RECORD
```

The next example leaves out records for Raleigh residents who own homes:

```
IF <CITY>=Raleigh  
AND <STATE>=North Carolina  
    OR=NC  
BUT NOT IF <RESIDENCE>=OWNED  
THEN PROCESS RECORD
```

You can use OR to modify BUT NOT IF. For example:

```
IF <CITY>=Pittsburgh  
AND <STATE>=Pennsylvania  
    OR=PA  
BUT NOT IF <RESIDENCE>=OWNED  
    OR=TEMPORARY  
THEN PROCESS RECORD
```

NOTE: *If you put AND after BUT NOT IF, AND modifies the IF at the top of the IF statement. If you use two successive BUT NOT IFs, both modify the IF at the beginning of the IF statement.*

Using More Than One IF Statement

To use more than one IF statement in a list specification document, link IF statements with OR.

The next list specification document consists of three IF statements. Blank lines between IF statements show the proper grouping of ideas:

```
IF <STATE>=Arizona
THEN PROCESS RECORD

OR IF <STATE>=California
    AND <COUNTY>=Orange
THEN PROCESS RECORD

OR IF <STATE>=New Mexico
    AND <COUNTY>=Bernalillo
        OR=Santa Fe
        OR=Quay
THEN PROCESS RECORD
```

Each record that satisfies at least one IF statement is selected. The list specification document selects records that pertain to Arizona and to counties in two other states.

Testing for Empty Fields

If only a RETURN follows the equals sign in an IF statement, List Processing selects only the records that have no values for the field you identify. By testing for a blank field value, you can locate records you need to update.

To select records with empty field values:

Press RETURN immediately after the equals sign in an IF statement.

The next list specification document selects records with an empty <ZIP> field value:

```
IF <ZIP>=(RETURN)
THEN PROCESS RECORD
```

You may need separate list specification documents to select records. Suppose your records have an <HOURLY> field that contains a YES or NO value. With one list specification document, you can select the YES records. With another, you can select the NO records. With a third, you can select records with an empty <HOURLY> field value.

***NOTE:** List Processing does not select records that are missing the field name given in your IF statement. If List Processing does not find the specified field name in a record, the record is not processed.*

Testing with Wild Cards

Wild cards are symbols that tell List Processing to substitute any characters for the wild card symbols. Wild cards are useful when you want to select all records that have any character or characters in a certain field.

List Processing has two wild cards, <?> and <*>. The <?> tests for any single character. The <*> looks for more than one character.

The <?> (question mark) wild card. To test for any single character, use the <?> wild card. The next list specification document tells List Processing to select all records whose <CREDIT> fields contain exactly one character. Records with <CREDIT> field values that are empty or missing, or records with more than one character, are not selected:

```
IF <CREDIT>=<?>
THEN PROCESS RECORD
```

You can combine the <?> wild card with partial field values. List Processing looks for the characters you specify and for any character where you used <?>. For example:

```
IF <MODEL#>=R<?>-15<?>
THEN PROCESS RECORD
```

List Processing looks in <MODEL#> fields for a capital R, any single character, a hyphen, a 1, a 5, and then any character, in that order. For example, RA-156 and R3-15B are two model numbers that are selected for processing.

To look for a certain number of characters, which can be any characters at all, type one question mark in the angle brackets for each character.

The next list specification document searches for field values with exactly five characters at the start, before the letters *boro*:

```
IF <TOWN>=<?????>boro  
THEN PROCESS RECORD
```

Fields named <TOWN> with values Northboro and Southboro match this selection. Westboro (only four characters before *boro*) does not.

The <*> (asterisk) wild card. To look for any number of characters, which can be any characters at all, including none, use the <*> wild card. In the next example, any <TOWN> field value matches this specification if the last four characters are *boro*. Missing and empty field values do not match:

```
IF <TOWN>=<*>boro  
THEN PROCESS RECORD
```

Values such as Northboro, Southboro, Eastboro, and Westboro match. A field containing only the four characters *boro* also matches. Records with no field value after <TOWN> are not selected.

The next list specification document selects any record where the <NAME> field value contains Jones, regardless of the characters that precede or follow it:

```
IF <NAME>=<*>Jones<*>  
THEN PROCESS RECORD
```

Values such as Jones, Mary Jones, J.P. Jones, Sr., or John Paul Jones, Jr., all match this list specification.

The next list specification document selects any record that contains a <ZIP> field name, even if the record does not contain a <ZIP> field value:

```
IF <ZIP>=<*>  
THEN PROCESS RECORD
```

Testing Numeric Fields

There are two types of fields: numeric and nonnumeric. Nonnumeric fields have the format <FIELD NAME>. Nonnumeric field values can contain both letters and numbers. Numeric fields have a colon after the left angle bracket, <:FIELD NAME>. Numeric field values contain numbers, but can include a dollar sign (\$), commas (,), a decimal point (.), an asterisk (*), and a plus sign (+), minus sign (-), or parentheses.

A numeric field is useful when you want to select records with field values containing both numbers and a dollar sign, comma, decimal point, asterisk, plus or minus sign, or parentheses. For example:

```
IF <:COST>=<10000>
THEN PROCESS RECORD
```

This specification selects records with field values 10,000, 10,000.00, 10000, and \$10,000. It would not select \$1,000, -10,000, (10,000) or AA-10000. List Processing tests the numeric value of a field if:

- The field name is numeric, such as <:COST>
- The field value is in angle brackets, such as: <10,000>

When you test the value of a number, List Processing:

- Ignores leading zeros, asterisks, plus signs, commas, and dollar signs in the field value. Thus, 10000, 010000, *10000, +10000, 10,000, and \$10000 are all selected for processing.
- Interprets parentheses around a number, or leading or trailing minus signs, as indicating a negative value. Thus, (10000), -10000 and 10000- are equal values. But these numbers have a value different from 10000. Records with these values are not selected when the specification is IF <:COST>=<10000>.
- Allows for an implied decimal point, or a decimal point and up to six zeros. Thus 10000 and 10000.00 are selected for processing.

- Issues an error message and stops if letters or special characters (other than those mentioned above) are found in the list document field values. For example, the field values !10,000, 10/00/00, and A10000 cause error messages when `<:COST>=<10000>` is the specification.

Testing Character-for-Character

List Processing tests a field value character-for-character if the value on the right of the equals sign is not enclosed in angle brackets, such as `<:COST>=10000`. The field name can be numeric or nonnumeric.

Many of the examples in this chapter use character-for-character comparisons in the IF statement. Note that this method is an exact way of specifying the field value. Only those records with every character you specify for the field value are processed. For example, suppose you want to select records for projects that cost \$10,000:

```
IF <:COST>=$10,000  
THEN PROCESS RECORD
```

OR

```
IF <COST>=$10,000  
THEN PROCESS RECORD
```

List Processing looks for a character-for-character match. Records without the dollar sign or comma are not selected. Only records with the field value \$10,000 are selected.

Testing for Digits Only in a Nonnumeric Field

List Processing ignores all characters, except digits, in a field value when:

The field name is not numeric and the field value is in angle brackets, such as `<COST>=<100>`.

The next example shows how to test only the numbers in the field value:

```
IF <COST>=<10000>
THEN PROCESS RECORD
```

The field values \$10,000, \$10000, 10000, \$100.00, AA-10000, 10/0/00 and 1.0000 for <COST> match because each contains the digits 10000. The values 10 and 100,000 do not match because they contain too few or too many digits. You can add characters to the field value in the specification document. For example, IF <COST>=<10000> selects the same records as IF <COST>=<\$10,000>.

NOTE: You cannot use the <> and <?> wild cards on a line where you are using the angle bracket method to check for a number or numeric range.*

Testing Numeric Ranges

In any condition in an IF statement, you can test numeric fields for values within a range. List Processing compares the field values in the list document with the range in the IF statement.

To test for a range of values, use one of these formats:

```
IF <:FIELD NAME>=<FIELD VALUE> OR MORE
```

For example:

```
IF <:ITEM#>=<10> OR MORE
```

```
IF <:FIELD NAME>=<FIELD VALUE> OR LESS
```

For example:

```
IF <:COST>=<$9.00> OR LESS
```

```
IF <:FIELD NAME>=<FIELD VALUE 1> THROUGH <FIELD VALUE 2>
```

For example:

```
IF <:INVENTORY>=<51> THROUGH <100>
```

Make sure to put field values in angle brackets when you test with a range.

The next example shows how to select records for employees whose salaries fall in a range:

```
IF <:SALARY>=<10000> THROUGH <14999>  
THEN PROCESS RECORD
```

NOTE: You can use *THRU* for *THROUGH*.

To select customers who have owed \$50 or more for at least three months, use:

```
IF <:BALANCE>=<$50.00> OR MORE  
AND <:OVERDUE>=<3> OR MORE  
THEN PROCESS RECORD
```

Testing Dates

If you do not want to test dates against a range, you can use a nonnumeric field value to represent a date. If you want to have the option of testing dates against a range, see the next section.

If you are not going to use ranges to select dates, use these guidelines to test dates:

- Enter the month, day, and year as two-digit numbers in the list document. For example: <DATE>05/19/49.
- Test for numbers only in the list specification document. For example, the statement IF <DATE>=<*>/49 finds all records for the year 1949.
- Use a nonnumeric field name.

If you do not enter the date as a two-digit number, you might get errors in a digits-only comparison. For example, when you use IF <DATE> = <1280>, List Processing processes records with the dates 1/2/80 and 12/80. Records with the dates 01/2/80 and 01/12/80 are not processed. See the section Testing for Digits Only in a Nonnumeric Field.

Testing Dates Against a Range

To process records that have dates that fall within a range:

- 1 Use the format YYMMDD (year, month, day) in the list document.
For example: <:DATE>780621 represents June 21, 1978.

Do not include dashes or slashes since numeric fields cannot accept these characters.
- 2 Type angle brackets around the dates of the range. You can use either THRU or THROUGH.

For example:

```
IF <:DATE>=<410714> THROUGH <780621>  
THEN PROCESS RECORD
```

This list specification document tests for dates that fall between July 14, 1941 and June 21, 1978.

List Specification Guidelines

Follow these guidelines when you create a list specification document:

- Use as many characters or lines in the list specification document as you want.
- Use spaces or tabs to indent lines of a list specification document where desired. These spaces and tabs make no difference to the instructions, but make the instructions more readable.
- Type each field name exactly as it appears in list and form document records.

- Type each field value immediately after an equals sign.
- Put spaces or tabs in a field value only if they also appear in the field values in the list document. Uppercase letters match only uppercase letters. For example, the statement IF <CREDIT>=AA does not process records with the field <CREDIT>aa.

NOTE: If you highlight the field names in the form document, the field value will not be highlighted in the output document. You must highlight the field values in the list document if you want them highlighted in the output document.

- Put each field value or range on a separate line.
- Type the words THEN PROCESS RECORD on a separate line.

Testing a List Specification Document

List Processing tests the list specification document before processing. However, you can use the Test Specification Document (T) option on the List Processing Menu to test a complex list specification document before running List Processing. List Processing issues error messages for missing angle brackets, misspelled key words, such as THROUGH, and missing phrases.

To test a list specification document:

- 1 Enter T (Test Specification) at the List Processing Menu.
The Test Specification Document form appears.
- 2 Enter the name or number of the document you want to test. If necessary, include the drive and directory.
- 3 Press RETURN.

If there are no errors in the document, List Processing tells you with a screen message:

```
The specification document is correct
Press any key to continue
```

If the list specification document contains errors, the text that caused the first error is displayed with an error message.

To correct the error, edit your list specification document.

Words and Phrases in a List Specification: Summary

Figure 4-1 shows the words and phrases you can use in a list specification document. In Figure 4-1, (range) indicates you have the option of using a range. The ellipses indicate that the entire line may be repeated for other field values.

```

IF <FIELD NAME>=field value (range)

    OR=field value (range) ...

    AND <FIELD NAME>=field value (range)

        OR=field value (range) ...

        BUT NOT IF <FIELD NAME>=field value (range)

            OR=field value (range) ...

THEN PROCESS RECORD

    OR

IF ...
  
```

where range is one of these:

```

    THROUGH <field value>
    THRU <field value>
    OR MORE
    OR LESS
  
```

Figure 4-1 Words and Phrases in a List Specification

NOTE: *The field value before **THROUGH**, **OR MORE**, and **OR LESS**, and the field value after **THROUGH**, must be in angle brackets.*

Table 4-1 summarizes the words and phrases used in a list specification document. Use this table once you are familiar with List Processing.

Table 4-1 Words and Phrases in a List Specification

Use	To
AND field name = field value	Select records with test values in more than one field.
BUT NOT IF field name = field value	Reject records with a test value in a field among records you are selecting.
IF...THEN	Test if one or more fields in each list document record meets test conditions. IF...THEN PROCESS RECORD is an IF statement.
OR	Select records that satisfy at least one of two IF statements.
OR = field value	Select or reject records with at least one of two or more test values in certain fields.
OR LESS	Test a field for a value equal to or less than a test value. The field value before OR LESS must be in angle brackets.
OR MORE	Test a field for a value equal to or greater than a test value. The field value before OR MORE must be in angle brackets.
PROCESS RECORD	Process all records in a list document that meet the test conditions. If there is no IF...THEN, all records are processed.

(continued)

Table 4-1 Words and Phrases in a List Specification (Cont.)

Use	To
THROUGH field value	Test a field for a value equal to or greater than the preceding test value and less than or equal to the following test value. The field values before and after THROUGH must be in angle brackets.
<?>	Test for the presence of exactly one character, which can be any character, in a field value.
<*>	Test for any number of characters, which can be any characters at all, or no characters, in a field value.
<>, enclosing a field value	Test for a numeric value if the field name is numeric, or test for a sequence of digits if the field name is not numeric.

List Processing Math

With List Processing Math, you set up calculations to be performed on field values in list document records. List Processing Math performs the calculations and places the results in the output document.

If you change the field values and run List Processing again, List Processing Math calculates the new results.

This chapter explains how to use List Processing Math to:

- Perform addition, subtraction, multiplication, and division on numbers in list documents
- Format your results
- Check for errors in your formulas, format strings, and input data

List Processing Math and Editor Math

In many ways, List Processing Math is like Editor Math, which is documented in *WPS-PLUS Editor Functions*.

Table 5-1 summarizes the differences between Editor Math and List Processing Math.

Table 5-1 Editor Math and List Processing Math

Editor Math	List Processing Math
Starts calculations when you are using the editor and you move the cursor through a BEGIN control block.	Starts calculations during List Processing when a FORMULA control block is reached.
Gets information from decimal-aligned tab positions.	Gets information from input fields in the list document. Input fields are the fields identified on the left of equals signs in FORMULA commands.
Uses all possible formulas on every line of the table until another control block is reached.	Uses all possible formulas on every record of the list until another control block is reached.
Answers appear in the table.	Answers appear in the output document where indicated by result fields in the form document. A result field is a field identified on the right of an equals sign in a FORMULA command.
Reworks calculations whenever you scroll through the table while in the WPS-PLUS Editor.	Reworks calculations whenever you run List Processing with the list document.

Using List Processing Math

To use List Processing Math:

Insert List Processing Math commands within control blocks at the top of your list document, or between records in your list document.

A FORMULA control block contains a WPSMATH command and at least one FORMULA command. FORMULA commands tell List Processing Math which calculations to perform on field values in records after the control block.

Figure 5-1 shows a sample FORMULA command control block.

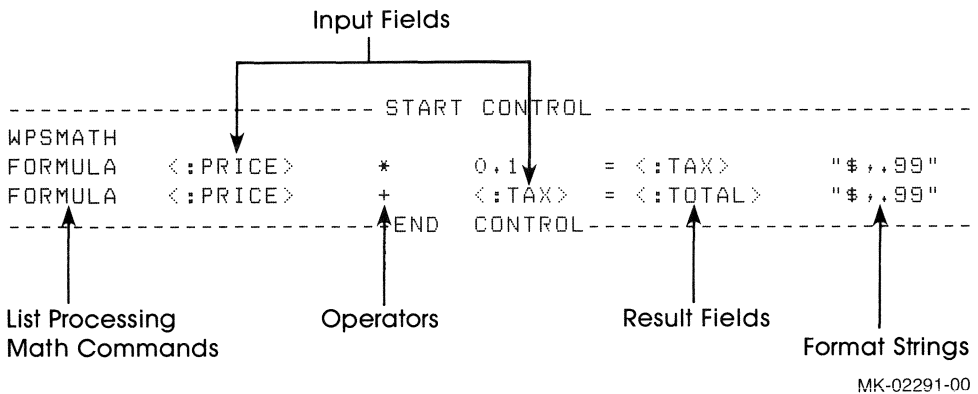


Figure 5-1 Sample FORMULA Command Control Block

This sample FORMULA command control block consists of five basic components:

- List Processing Math commands, which start List Processing Math and indicate the calculations to perform. The WPSMATH command starts List Processing Math. The two FORMULA commands contain calculations.
- Input fields, which always appear to the left of the equals sign in a FORMULA command. These field names and the corresponding field values are stored in the body of your list document.
- Operators, which tell List Processing Math whether to add, subtract, multiply, or divide.

- Result fields, which always appear to the right of the equals sign in a FORMULA command. Result fields tell List Processing Math where to store the results of the calculations. These field names may correspond to field names in your form document.
- Format strings, which are strings of characters at the end of a FORMULA command. Format strings tell List Processing Math how to format results.

Processing List Processing Math

When you run List Processing, List Processing Math:

- Does calculations in the order in which the formulas appear in a FORMULA control block.
- Applies every formula in the control block to the records that follow.

NOTE: List Processing Math works only on the selected records. If a list specification, or a setting on the List Processing Documents Selected form, excludes a record, List Processing Math does not use field values in the record for calculations.

- Gives you an error message if there is an error in your control block.

Press RETURN to return to the Document Processing Menu, then edit your list document to correct the error.

Starting List Processing Math

List Processing Math starts calculating when it comes to a FORMULA control block in the list document.

To perform calculations on all records in a list document:

Put FORMULA commands in a control block before the first list document record.

To perform calculations on certain records only:

- 1 Put FORMULA commands in a control block before the records to be used.
- 2 Put a null control block after the records.

See the section Stopping List Processing Math.

NOTE: You can also use your list specification document to select or exclude certain records for processing.

To change the calculations you are performing:

Insert another FORMULA control block where the new calculations start.

The earlier set of calculations stops at that point.

How to Create a FORMULA Control Block

To create a FORMULA control block in a list document:

- 1 Press Gold CMND to insert a START CONTROL marker.
- 2 Type WPSMATH.
- 3 Press RETURN.
- 4 Type FORMULA and press the space bar.
- 5 Type the formula you want List Processing Math to use.

A formula includes the names of input fields separated by the arithmetic operators, +, -, *, or /. Input fields are followed by an equals sign, and the name of the result field. For example:

```
FORMULA <:NUMBER OF ITEMS> * <:ITEM COST> = <:COST OF ITEMS>
```

Input fields must be numeric fields that appear in list document records since List Processing Math gets input data from the list document. A left angle bracket and colon indicate a numeric field name.

If your formula is too long for a single line, let it wrap to the next line.

- 6 Press the space bar.
- 7 Include a format string in quotation marks if you want results in a customized format.
- 8 Press RETURN.
- 9 Press Gold CMND to insert an END CONTROL marker.

Stopping List Processing Math

List Processing Math stops calculating when it:

- Encounters the end of the list document
- Encounters a null control block
- Processes the final record selected by your list specification document

A null control block contains a WPSMATH command but no FORMULA commands.

A null control block looks like this:

```
----- START CONTROL -----  
WPSMATH  
----- END CONTROL -----
```

To create a null control block:

- 1 Press Gold CMND to insert a START CONTROL marker.
- 2 Type WPSMATH.
- 3 Press RETURN.
- 4 Press Gold CMND to insert an END CONTROL marker.

List Processing Math starts calculating again if it comes to a FORMULA control block.

Hint. You can include SET commands in a null control block to set fields or store values. For example, you can insert this command in a null control block to initialize an accumulator: SET 0 = <:TOTAL COST>. For more information about the SET command, see *WPS-PLUS Editor Functions*.

List Processing Math Commands

The first component of a command control block is a List Processing Math command. Table 5-2 shows the five List Processing Math commands:

Table 5-2 List Processing Math Commands

Use	To
WPSMATH	Start List Processing Math. Without a FORMULA command in the same control block, the WPSMATH command stops List Processing Math.
FORMULA	Identify input fields, result fields, and the arithmetic operations you want List Processing Math to perform.
SET	Assign a value to a field.
TRUNCATE	Drop extra digits from a decimal fraction without rounding off. The number of digits in the fraction after truncation is the number specified in the format string, or at most six if there is no format string. Trailing zeros are dropped.
ROUND	Round off a decimal fraction. The fraction is rounded off to the number specified in the format string, or at most to six places, if there is no format string. Trailing zeros are dropped.

FORMULA Command Guidelines

FORMULA commands tell List Processing Math what calculations to perform.

When you write FORMULA commands, follow these guidelines:

- Only use numeric fields in FORMULA commands. A numeric field begins with a left angle bracket and colon, <:.
- Put input fields to the left of the equals sign, =, in FORMULA commands.
- Use the symbols, + - * and /, to tell List Processing Math which arithmetic operations to perform.
- Put result fields to the right of the equals sign.
- Put as many FORMULA commands as you need in a control block.

Allowable Characters

Besides field names, the only characters allowed in List Processing Math formulas are those shown in Table 5-3.

Table 5-3 Allowable Characters

Character	Description
0 1 2 3 4 5 6 7 8 9	Numerals, used to write constants. Constants are numbers you write into formulas. They do not change from one line of a table to the next, as field values usually do.
. ,	Decimal point and comma, which can be used to write constants.
\$	A dollar sign, indicating cash value.
+	A plus sign, indicating addition or a positive number.
-	A minus sign, indicating subtraction or a negative number.
*	An asterisk, indicating multiplication.
/	A slash, indicating division.

(continued)

Table 5-3 Allowable Characters (Cont.)

Character	Description
()	Parentheses, used to group operations together. The parentheses must always match. For every right parenthesis,), there must be exactly one left parenthesis, (.
=	An equals sign, indicating the end of the calculation. Every formula must contain only one equals sign. Do not put + - * / (or) on the right side of an equals sign. Instead, you must follow the equals sign with the name of a field where you want the answer to go.
	Spaces and tabs. You can put spaces or tabs in a FORMULA command. List Processing Math ignores them, but they can make formulas easier to read. Do not put extra spaces in a field name, or List Processing Math does not recognize the name.

Constants

A constant is any number in a formula. For example:

```
FORMULA <:PRICE> * 0.1 = <:TAX>
```

In this example, 0.1 is a constant.

You can add or subtract a constant, or multiply or divide by a constant. A constant can be negative or positive, a whole number, or a fraction.

List Processing Math: Complete Example

Figure 5-2 shows a list document with List Processing Math commands, a sample form document, and a sample output document.

The list document:

This is a list of our sales force, their sales for the month, and their expenses. Calculated data will include sales commission and Profit.

```
----- START CONTROL -----  
WPSMATH  
FORMULA <:SALES> * 0.1 = <:COMMISSION> "$99,999.99"  
FORMULA <:SALES> - <:COMMISSION> - <:EXPENSES> = <:PROFIT> "$99,999.99"  
----- END CONTROL -----  
<NAME>Joan Hamett  
<AREA>Boston  
<:SALES>$8809.00  
<:EXPENSES>$550.78  
<>  
  
<NAME>Sara Hayes  
<AREA>New York  
<:SALES>$14,238.67  
<:EXPENSES>$924.62  
<>
```

The form document:

```
Sales Representative Report  
  
<IS>  
<NAME>, working in <AREA>, made <:SALES>. Expenses were  
<:EXPENSES> and the commission was <:COMMISSION>, leaving a Profit of  
<:PROFIT>.  
<IE>
```

The list specification document:

PROCESS RECORD

The output document:

```
Sales Representative Report  
  
Joan Hamett, working in Boston, made $8809.00. Expenses were  
$550.78 and the commission was $880.90, leaving a Profit of  
$7,377.32  
  
Sara Hayes, working in New York, made $14,238.67. Expenses were  
$924.62 and the commission was $1,423.87, leaving a Profit of  
$11,890.18.
```

MK-02290-00

Figure 5-2 List Processing Math: Complete Example

The first FORMULA command multiplies the total sales for each sales representative by 0.1 to produce the commission.

The second FORMULA command subtracts the commission and expenses for each representative from the sales to produce the profit.

The form document indicates where the results go in the output document. The result fields are <:COMMISSION> and <:PROFIT>.

The list specification document tells List Processing Math to process all records in the list document.

The sample output shown is part of a sales report. The sales report contains field values from the list document and results calculated by List Processing Math.

Averages

When calculating averages, you can use the special accumulator <:LINES> to total the number of records processed.

To total the number of records processed:

Put the accumulator <:LINES> in the FORMULA control block at the top of the list document.

<:LINES> counts the number of list document records processed. Divide by the value in <:LINES> to produce the average over the records processed.

For example, suppose you need to find the average value of an inventory item. These commands calculate the totals you need to find the average:

```
----- START CONTROL -----  
WPSMATH  
FORMULA <:LINES> + 1 = <:LINES>  
FORMULA <:TOTAL> + <:COST> = <:TOTAL> "$,99"  
----- END CONTROL -----
```

The first formula increases the value of <:LINES> by one with each record processed. Initially, the value for <:LINES> is zero.

The second formula calculates the total cost of your inventory items by adding the cost in each record to <:TOTAL>.

To find the average cost of an inventory item, divide <:TOTAL> by <:LINES>:

```
FORMULA <:TOTAL> / <:LINES> = <:AVERAGE> "$,,99"
```

To see the values in <:TOTAL>, <:LINES>, or <:AVERAGE>, put the field name in the trailer of the form document.

Here is the list document with commands included:

```
----- START CONTROL -----
WPSMATH
FORMULA <:LINES> + 1 = <:LINES>
FORMULA <:TOTAL> + <:COST> = <:TOTAL> "$,,99"
FORMULA <:TOTAL> / <:LINES> = <:AVERAGE> "$,,99"
----- END CONTROL -----

<:PART NUMBER>01
<PART NAME>Binder
<:COST>$1.19
<:NUMBER IN STOCK>47
<>

<:PART NUMBER>02
<PART NAME>Cellophane tape
<:COST>$1.59
<:NUMBER IN STOCK>97
<>
```

Here is a form document with a trailer that shows the average cost of inventory items:

INVENTORY REPORT: 11-5-84

<!S>

<:PART NUMBER> <PART NAME>

<:NUMBER IN STOCK> in stock @ <:COST>

<!E>

Inventory includes <:LINES> parts, at an average cost of <:AVERAGE> each.

The list specification document includes only the words PROCESS RECORD.

Here is a sample output document:

INVENTORY REPORT: 11-5-84

01 Binder

47 in stock @\$1.19

02 Cellophane tape

97 in stock @\$1.59

Inventory includes 2 parts, at an average cost of \$1.39 each.

Totals and Subtotals

To calculate totals and subtotals:

Use FORMULA commands with accumulator fields.

Accumulator fields store running totals.

For example, suppose you need to total the cost of your inventory. Here is a sample list document:

```
----- START CONTROL -----
WPSMATH
FORMULA <:COST> * <:NUMBER IN STOCK> = <:INVENTORY COST> "$,,99"
FORMULA <:TOTAL COST> + <:INVENTORY COST> = <:TOTAL COST> "$,,99"
----- END CONTROL -----
<:PART NUMBER>01
<PART NAME>Binder
<:COST>$1.19
<:NUMBER IN STOCK>47
<>

<:PART NUMBER>02
<PART NAME>Cellophane tape
<:COST>$1.59
<:NUMBER IN STOCK>97
<>
```

The first command in the control block calculates the in-stock value of an inventory item.

The second command adds the in-stock value of each inventory item to the running total in the accumulator field <:TOTAL COST>.

NOTE: *If you used an accumulator before in a list document, you must set the accumulator to zero before starting a total or subtotal. Initialize the accumulator by inserting a SET command in a null control block. For example, SET 0 = <:TOTAL COST>.*

To display a total in your output document:

Include the accumulator in your form document, in the trailer, for example.

The next example shows a way to calculate subtotals. Suppose sales representatives are sorted into Northern and Southern regions in your list document. You want to subtotal the sales for each region and also keep a running total of all sales. Your list document contains sales representatives in the Northern region, followed by sales representatives in the Southern region.

Figure 5-3 shows the list document with List Processing Math commands.

List Processing Math

SALES COMMISSIONS REPORT:

THIS LIST DOCUMENT CONTAINS RECORDS OF SALES COMMISSIONS IN BOTH REGIONS,
NORTH AND SOUTH.

```
----- START CONTROL -----  
WPSMATH  
FORMULA <:NORTH> + <:COMMISSION> = <:NORTH> "$*,99999.99"  
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "$*,99999.99"  
----- END CONTROL -----
```

NORTH DISTRICT:

```
<LAST>Tucker  
<FIRST>Fred  
<:EMPLOYEE NUMBER>001  
<:COMMISSION>$13,880.25  
<>  
<LAST>Bascomb  
<FIRST>Roger  
<:EMPLOYEE NUMBER>002  
<:COMMISSION>$13,009.00  
<>  
<LAST>Robertson  
<FIRST>Michael  
<:EMPLOYEE NUMBER>003  
<:COMMISSION>$10,400.00  
<>
```

```
----- START CONTROL -----  
WPSMATH  
FORMULA <:SOUTH> + <:COMMISSION> = <:SOUTH> "$*,99999.99"  
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "$*,99999.99"  
----- END CONTROL -----
```

SOUTH DISTRICT:

```
<LAST>Whitcomb  
<FIRST>David  
<:EMPLOYEE NUMBER>004  
<:COMMISSION>$14,000.90  
<>  
<LAST>Stuart  
<FIRST>Roger  
<:EMPLOYEE NUMBER>005  
<:COMMISSION>$15,555.00  
<>
```

Figure 5-3 Sample List Document Showing Subtotals

In Figure 5-3, the command that calculates the subtotal for the Northern region is:

```
FORMULA <:NORTH> + <:COMMISSION> = <:NORTH> "$*,99999.99"
```

The command that calculates the total commissions is:

```
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "$*,99999.99"
```

A second WPSMATH control block starts subtotalling the Southern region.

The command that calculates the subtotal for the Southern region is:

```
FORMULA <:SOUTH> + <:COMMISSION> = <:SOUTH> "$*,99999.99"
```

When List Processing encounters the second control block, the first set of calculations stops. The second control block restarts the calculation for the grand total by repeating a command from the first control block.

Here is the form document.

```
COMMISSIONS EARNED IN ALL DISTRICTS (Q4)
```

```
<!S>
```

```
Employee <:EMPLOYEE NUMBER>: <:COMMISSION><!E>
```

```
SUMMARY:
```

```
    NORTH District employees: <:NORTH>
```

```
    SOUTH District employees: <:SOUTH>
```

```
    TOTAL commissions: <:GRAND TOTAL>
```

The trailer in the form document contains the Northern and Southern subtotals and the grand total.

The list specification document contains only the words PROCESS RECORD.

Here is the output document:

COMMISSIONS EARNED IN ALL DISTRICTS (Q4)

```
Employee 001: $13,880.25
Employee 002: $13,009.00
Employee 003: $10,400.00
Employee 004: $14,000.90
Employee 005: $15,555.00
```

SUMMARY:

```
    NORTH District employees: $37,289.25
    SOUTH District employees: $29,555.90

    TOTAL commissions:           $66,845.15
```

Truncation and Rounding

By default List Processing Math rounds off the decimal digit corresponding to the last 9 in a format string.

To truncate results instead of rounding off:

Use the TRUNCATE command.

Truncation simply drops extra digits. For example, if you use the TRUNCATE command and specify a format string of “.99” for the result 0.796, List Processing Math displays the result as 0.79.

Type the TRUNCATE command in a control block on a line by itself on any line after the word WPSMATH. For example:

```
----- START CONTROL -----  
WPSMATH  
FORMULA   <:COST> / 10 = <:REDUCTION> "$,9.99"  
TRUNCATE  
----- END   CONTROL -----
```

To resume rounding off:

Use a ROUND command.

Type ROUND on a line by itself in a control block.

Always give a separate line to a TRUNCATE or ROUND command. Specify only one TRUNCATE or ROUND command per control block.

Default Format

Unless you include a format string in a FORMULA command, List Processing Math writes results in default format. These are the characteristics of default format:

- Dollar signs, plus signs, and commas are not included.
- If a number is negative, a minus sign is in front of the number.
- If the number is not a whole number, the number contains a decimal point and digits for the fractional part.
- No zeros are displayed after the last nonzero digit of a decimal fraction.
- Numbers can have a maximum of thirteen digits. Up to six of those digits can follow the decimal point.
- Only the first six digits of fractions are displayed.

Use format strings when you want results in a customized format, such as with dollar signs (\$), or rounded off to a certain number of decimal places.

Customized Format

To write results in a different format:

Put a format string in quotation marks at the end of your FORMULA command.

The characters you can include in a format string are $-$ $+$ $($ $)$ $\$$ $*$ $.$ and 9 .

For example, the format string in the following command tells List Processing Math to round off its answer to one decimal digit:

```
FORMULA <:IN STOCK> + <:NEW> = <:IN STOCK> ",9"
```

Decimal Fractions

To format results containing decimal fractions:

Decide on the largest number of digits you need right of the decimal point, then use that number of 9s after the decimal point in your format string.

You can drop some decimal digits by including fewer 9s after the decimal point than the full answer requires. You can include at most six 9s to the right of the decimal point.

For example:

```
FORMULA <:COST> * <:ITEMS> = <:TOTAL> ".999"
```

The format string in this example tells List Processing Math to round off the result to three decimal places.

In default format, decimal fractions are written with one zero to the left of the decimal point. For example, the default format for one half is 0.5. To avoid displaying the initial zero, specify the format string ".9", which causes one half to be written as .5.

To drop all decimal digits and round answers to the nearest integer:

Use the format “9.” (to print the decimal point), or “9” (to avoid printing the decimal point).

You do not need to include 9s left of the decimal point unless your format string contains an asterisk. See the section Asterisks.

Commas

If your results are large numbers, use commas to group digits for easier reading. In a format string, a comma must come after a dollar sign and before a decimal point. One comma is all you need.

The following strings give the same results:

"9,999,999.99" ",.99"

Dollar Signs

If you include the dollar sign (\$) at the beginning of your format string, List Processing puts a dollar sign in front of the answer. To format results in dollars and cents, use the format string:

"\$,.99"

Trailing Minus Sign

A trailing minus sign in a format string tells List Processing Math to put the minus sign after a negative number instead of before it (for example, 775.01–). In a format string, a trailing minus sign must be the last symbol before the closing quotation mark. Examples of the trailing minus format are:

"*9999,99–" "*999,999–" "\$, .99–"

Leading Plus Sign

A leading plus sign in a format string tells List Processing Math to put a plus sign before positive numbers. An example of the leading plus sign format is:

```
" +999.99 "
```

Parentheses

Parentheses in a format string tell List Processing Math to write negative numbers in parentheses instead of with a minus sign. Here is a negative result written with parentheses:

```
( 999.01 )
```

Make parentheses the first and last symbols in a format string:

```
" ( $999.99 ) "
```

```
" ( ) "
```

```
" ( * ,999999 ) "
```

Notice that you do not have to include 9s to use parentheses format. If you include just “()” as your format string, List Processing Math writes its answers in default format, except that it uses parentheses instead of a leading minus sign to indicate negative numbers.

Asterisks

You can fill out results with asterisks. That way, you can keep a column of results even on the right. For example:

```
$84,000.00
```

```
$*4,343.00
```

```
$***778.00
```

To fill out a result with asterisks:

- 1 Put one asterisk after any dollar sign and before any commas or 9s.
- 2 Put in as many 9s left of the decimal point as there are digits and commas in the largest result you expect.

List Processing writes results so the number of asterisks, commas, and numerals left of the decimal point is the same as the number of 9s in the format string.

You can put a maximum of thirteen 9s in a format string, including a maximum of six right of the decimal point.

For example:

```
FORMULA <:COST> * <:ITEMS> = <:TOTAL> "$*999,999,999.99"  
FORMULA <:COST> * <:ITEMS> = <:TOTAL> "$*,999999999.99"
```

The format strings in these commands produce the same results. If a result contains fewer than nine digits and commas left of the decimal point, List Processing Math adds asterisks until the total of digits, commas, and asterisks is nine. For example:

```
$***94,000.00
```

If a result contains nine digits and commas left of the decimal point, no asterisks are added. For example:

```
$4,320,213.00
```

List Processing Math Error Messages

When List Processing Math finds an error, processing stops and an error message displays. Part of the line containing the error is displayed above the message. The line ends at the point where List Processing Math could not understand the command or the number. The location of the error is marked with a caret, ^.

For example:

```
FORMULA <:CREDIT> + <:TOTAL> <:TOTAL>  
-----^  
Formula error
```

In this example, an equals sign was omitted from a FORMULA command. Notice that the error is not always at the point where List Processing Math ends the line. Sometimes, you can correct the error by changing something earlier in the line.

Most errors occur because List Processing Math cannot understand what your commands mean or what numbers you want to use. Use the following checklists to avoid and correct errors.

FORMULA Command Checklist

To avoid making formula notation errors, or to correct them, make sure that:

- The characters + - * / and = are always followed by field names or constants.
- All field names and constants are separated from their neighbors by + - * / or =.
- There is exactly one right parenthesis,), for every left parenthesis, (.
- There is a RETURN at the end of every formula, or, if there is a format string, at the end of the format string.
- There is exactly one equals sign, =, in every FORMULA and SET command, with exactly one field name after it.
- The commands WPSMATH, TRUNCATE, and ROUND each have a line to themselves.
- Every field name starts with <: and ends with >.
- The only characters in constants are 0 1 2 3 4 5 6 7 8 9 , + - . and \$.
- The characters + - . and \$ appear no more than once in any constant, in the correct places.
- No constant has more than thirteen digits, or more than six digits to the right of the decimal point.
- The command is complete. Make sure there is an equals sign, =, a constant, or a complete field name to the left of the equals sign, and a complete field name to the right of the equals sign.

Format Checklist

To avoid making errors and to correct errors in format strings, make sure that:

- Each format string is enclosed in quotation marks.
- The only characters in format strings are $+$ $-$ $($ $)$ $\$$ $*$ $,$ $.$ and 9 .
- The characters $+$ $-$ $($ $)$ $\$$ and $.$ appear no more than once.
- The plus sign, $+$, is first, if it appears.
- The minus sign, $-$, is either first or last, if it appears.
- The plus sign, $+$, and minus sign, $-$, do not appear in the same format string.
- Parentheses, $($ $)$, and the plus sign, $+$, do not appear in the same format string.
- Parentheses, $($ $)$, and the minus sign, $-$, do not appear in the same format string.
- There is exactly one right parenthesis, $)$, for every left parenthesis, $($.
- The dollar sign, $\$$, comes before everything except the plus sign, $+$, minus sign, $-$, or left parenthesis, $($.
- The asterisk, $*$, comes before 9 $,$ $.$ $)$ or a trailing minus sign, $-$.
- There are no commas after the decimal point.
- There are no more than thirteen 9s in the format string and no more than six 9s to the right of the decimal point.
- Format strings containing the characters $+$ $*$ $-$ $.$ always contain 9s.

Input Field Value Checklist

To avoid making errors and to correct errors involving values of input fields (in your list document), make sure that:

- There are no empty pairs of parentheses, $()$.
- The only characters in the field value are 0 1 2 3 4 5 6 7 8 9 $\$$ $*$ $-$ $+$ $,$ $.$ (and).

- The characters \$. + - (and) appear no more than once.
- The characters \$ * + and (appear before any digits.
- The plus sign, +, comes first, if it appears.
- The minus sign, -, is either first or last, if it appears.
- The plus sign, +, and minus sign, -, do not appear in the same field value.
- The dollar sign, \$, comes before any asterisk, *, and after the plus sign, +, or the minus sign, -.
- There is exactly one right parenthesis,), for every left parenthesis, (.
- Parentheses come first and last, if they appear.
- Parentheses and the minus sign, -, do not appear in the same field value.
- Parentheses and the plus sign, +, do not appear in the same field value.
- There are no more than thirteen digits in the field value, and no more than six digits to the right of the decimal point.

6

Sorting a List Document

Sort Processing arranges lists of alphabetic or numeric information in ascending or descending order. You create a sort specification document to tell List Processing how to sort the list. You use the sorted list as input for List Processing.

This chapter explains how to create sort specification documents to sort on one or more fields. In addition, the chapter contains:

- Sample output from Sort Processing
- The sort alphabet (which gives the order used in sorting)

Using Sort Processing

To use Sort Processing, you create:

- A list document

The list document contains the records to be sorted. You can sort a list document you have created for List Processing.

- A sort specification document

The sort specification document tells Sort Processing which record field to use in sorting records, and which order to sort in (ascending alphabetic/numeric order or descending alphabetic/numeric order).

When you run Sort Processing, you produce a sort output document. The sort output document is a sorted list.

Sort Specification Documents

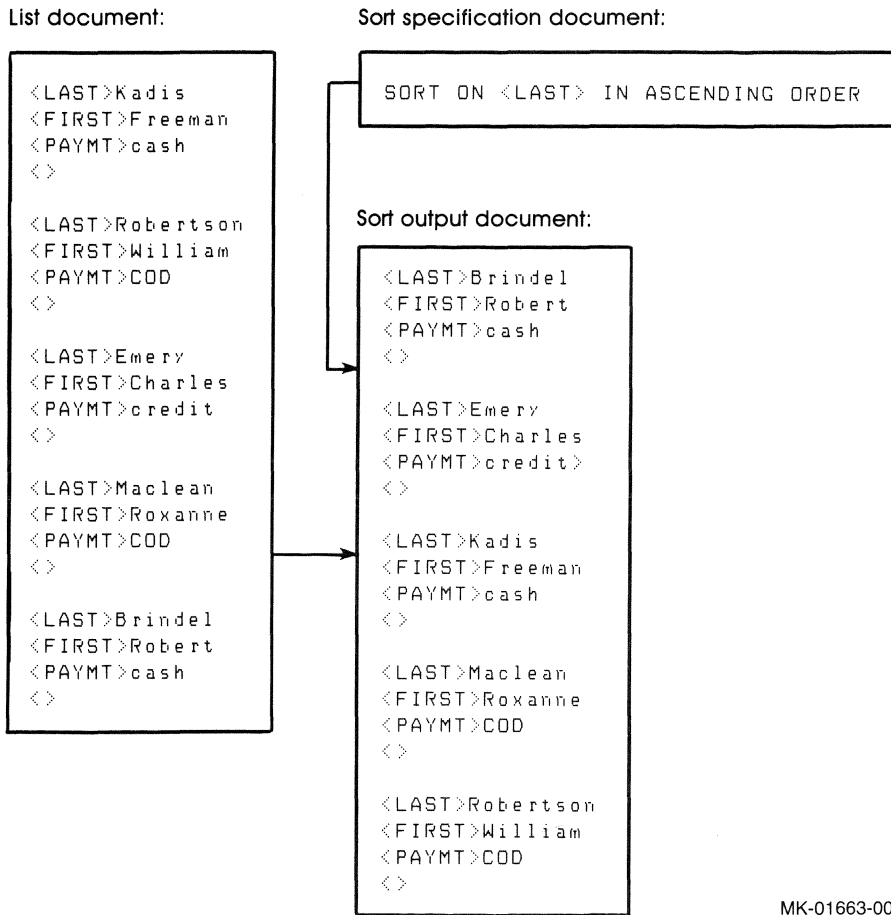
A sort specification document tells List Processing how to sort records in a list document.

You can sort records by one or more fields. For example, you can sort a mailing list by state so records for Alabama come first, followed by records for Alaska, Arizona, Arkansas, and so on. You can sort a mailing list by state and city so records for Allentown, Pennsylvania are grouped together, followed by records for Altoona, Pennsylvania, followed by those for Bethlehem, Pennsylvania and so on.

You can sort records in ascending alphabetic/numeric order or in descending alphabetic/numeric order.

How Sort Processing Works

Figure 6-1 shows how Sort Processing works.



MK-01663-00

Figure 6-1 How Sort Processing Works

In Figure 6-1, the sort specification document indicates that records are to be sorted by the <LAST> field, which contains last names, and in ascending order. Ascending order is the same as alphabetical order when letters are compared. Thus, Sort Processing arranges the list document records in alphabetical order by the last names in the records.

How to Create a Sort Specification Document

To create a sort specification document:

- 1 Enter C (Create) from the Document Processing Menu and fill out the Create form.

When you press RETURN, the screen clears and you are in the editor.

- 2 Type SORT ON and a space.
- 3 Type the name of the field you are using to sort records and press the space bar.
- 4 Type IN ASCENDING ORDER to sort in ascending alphabetic/numeric order.

To sort in descending order, type IN DESCENDING ORDER.

- 5 Press Gold FILE DOCMT to file the sort specification document.

How to Run Sort Processing

To run Sort Processing:

- 1 Enter SR (Sort Processing) from the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).

The Sort Processing Menu displays.

- 2 Enter SR (Sort Processing) from the Sort Processing Menu.

A form displays.

- 3 Follow the instructions at the bottom of your screen to run Sort Processing.

Sorting on More Than One Field

You can use as many fields as you need to use in sorting.

Here is a sample sort specification:

```
SORT ON <STATE> IN ASCENDING ORDER AND  
SORT ON <CITY> IN ASCENDING ORDER AND  
SORT ON <STREET> IN ASCENDING ORDER
```

This sort specification sorts list document records by the fields <STATE>, <CITY>, and <STREET>. Sort Processing organizes records by states, and within states, by cities, and within cities, by street addresses.

To sort on more than one field:

- 1 Put the primary sort field on the first line of the sort specification document.

The primary field is usually the largest classification among the fields sorted.

- 2 Specify the sort order for the primary field, ASCENDING or DESCENDING.

- 3 End the line with AND.

AND tells Sort Processing that another line follows.

- 4 Put other sort fields on subsequent lines of the sort specification document, arranged from largest to smallest classification, and specify their sort orders. End each line except the last line with AND.

In the previous example, the largest classification is <STATE>, so <STATE> is the primary sort field, the field used first in sorting records. The next largest classification is <CITY>, so <CITY> is the secondary sort field. The smallest classification is <STREET>, so <STREET> is the last sort field.

Figure 6-2 shows sample output.

```
<LAST>Baines
<FIRST>Louis
<STREET#>432
<STREET>Drayton Avenue
<CITY>Anaheim
<STATE>CA
<ZIP>92803
<>

<LAST>Whipple
<FIRST>Roser
<STREET#>74
<STREET>Main Street
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Hardy
<FIRST>Ted
<STREET#>900
<STREET>Thomason Avenue
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Billingham
<FIRST>Matthew
<STREET#>453
<STREET>Orchard Road
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>
```

Figure 6-2 List Sorted on <STATE>, <CITY>, and <STREET> in Ascending Order

In Figure 6-2, California records occur before New Hampshire records. Within California records, Anaheim records occur before Campbell records. Within Campbell records, Main Street records occur before Thomason Avenue records.

Examples of Sort Output Documents

The next example shows a sort in ascending order on two fields, <STATE> and <LAST>. Since <STATE> is the primary field, Sort Processing sorts by the <STATE> field first. Then Sort Processing sorts by the <LAST> field. The sort specification document is:

```
SORT ON <STATE> IN ASCENDING ORDER AND  
SORT ON <LAST> IN ASCENDING ORDER
```

Figure 6-3 shows the sort output document.

<LAST>Hutchinson
<FIRST>William
<ADD>3708 Loam Avenue
<CITY>Anaheim
<STATE>CA
<ZIP>92803
<>

<LAST>Robertson
<FIRST>William
<ADD>910 Hedge Road
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Briand
<FIRST>Robert
<ADD>159 Colby Street
<CITY>Nashua
<STATE>NH
<ZIP>03063
<>

<LAST>Franklin
<FIRST>Samuel
<ADD>608 Knoll Road
<CITY>Manchester
<STATE>NH
<ZIP>03101
<>

<LAST>Wilson
<FIRST>Sarah
<ADD>175 Inwood Avenue
<CITY>Manchester
<STATE>NH
<ZIP>03101
<>

Figure 6-3 List Sorted on <STATE> and <LAST> in Ascending Order

The next example shows a sort on the field name <ITEM#>. Some field values in the <ITEM#> field start with letters, others with numbers, still others with symbols or spaces. (Refer to Table 6-1 to see how sort places special symbols and numbers in ascending order.) The sort specification document is:

```
SORT ON <ITEM#> IN ASCENDING ORDER
```

Figure 6-4 shows the sort output document.

Sorting a List Document

```
<ITEM#>0008
<DESCRIP>DECmat 4'x6'
<SHIPPING WT>10 lbs.
<:UNIT PRICE>125.00
<>

<ITEM#>0074
<DESCRIP>Mylar Ribbon
<SHIPPING WT>1 lb.
<:UNIT PRICE>42.00
<>

<ITEM#>3009
<DESCRIP>WP rulers - pkg. of 10
<SHIPPING WT>.05 lbs
<:UNIT PRICE>$20.00
<>

<ITEM#>8976
<DESCRIP>Serial Interface Switch
<SHIPPING WT>2 lbs.
<:UNIT PRICE>$235.00
<>

<ITEM#>A0076
<DESCRIP>Acoustic cover
<SHIPPING WT>40 lbs.
<:UNIT PRICE>425.00
<>

<ITEM#>T0074
<DESCRIP>Fanfold Paper
<SHIPPING WT>27 lbs.
<:UNIT PRICE>43.00
<>
```

Figure 6-4 List Sorted on <ITEM#> in Ascending Order

The next example shows a sort on two fields. The primary field is <PAYMT> and the sort is descending. The secondary field is <LAST> and the sort is ascending.

The sort specification document is:

```
SORT ON <PAYMT> IN DESCENDING ORDER AND  
SORT ON <LAST> IN ASCENDING ORDER
```

Figure 6-5 shows the sort output document.

```
<LAST>Emery  
<FIRST>Charles  
<PAYMT>credit  
<>  
  
<LAST>Maclean  
<FIRST>Roxanne  
<PAYMT>COD  
<>  
  
<LAST>Robertson  
<FIRST>William  
<PAYMT>COD  
<>  
  
<LAST>Brindel  
<FIRST>Robert  
<PAYMT>cash  
<>  
  
<LAST>Kadis  
<FIRST>Freeman  
<PAYMT>cash  
<>  
  
<LAST>Spell  
<FIRST>Sarah  
<PAYMT> credit  
<>
```

Figure 6-5 List Sorted on <PAYMT> Descending and <LAST> Ascending

Notice that the last record contains a field value that begins with an accidental space. The space causes the record to be sorted last. Spaces are first in the sort alphabet. They are placed last when the sort is descending. Misplaced tabs can also affect your sorting.

The next example shows a sort on a numeric field. The list document is arranged in descending order according to the numeric value of the <:UNIT PRICE> field.

The sort specification document is:

```
SORT ON <:UNIT PRICE> IN DESCENDING ORDER
```

Figure 6-6 shows the sort output document.

```
<ITEM#>A0076
<DESCRIP>Acoustic cover
<SHIPPING WT>40 lbs.
<:UNIT PRICE>425.00
<COD>Y
<>

<ITEM#>8976
<DESCRIP>Serial Interface Switch
<SHIPPING WT>2 lbs.
<:UNIT PRICE>$235.00
<COD>Y
<>

<ITEM#>/008
<DESCRIP>DECmat 4'x6'
<SHIPPING WT>10 lbs.
<:UNIT PRICE>125.00
<CREDIT>Y
<>

<ITEM#>T0074
<DESCRIP>Fanfold Paper
<SHIPPING WT>27 lbs.
<:UNIT PRICE>43.00
<CREDIT>Y
<>

<ITEM#>0074
<DESCRIP>Mylar Ribbon
<SHIPPING WT>1 lb.
<:UNIT PRICE>42.00
<CASH>Y
<>

<ITEM#>3009
<DESCRIP>WP rulers - PKg. of 10
<SHIPPING WT>.05 lbs
<:UNIT PRICE>$20.00
<CASH>Y
<>
```

Figure 6-6 List Sorted on <:UNIT PRICE> in Descending Order

Sort Alphabet

Table 6-1 shows the sort alphabet, which is the sequence of symbols, numbers, and letters used by Sort Processing.

Table 6-1 Sort Alphabet

Sort Value	Character	Sort Value	Character	Sort Value	Character
1	space	23	6	45	L l
2	!	24	7	46	M m
3	"	25	8	47	N n
4	#	26	9	48	O o
5	\$	27	:	49	P p
6	%	28	;	50	Q q
7	&	29	<	51	R r
8	'	30	=	52	S s
9	(	31	>	53	T t
10	)	32	?	54	U u
11	*	33	@	55	V v
12	+	34	A a	56	W w
13	,	35	B b	57	X x
14	-	36	C c	58	Y y
15	.	37	D d	59	Z z
16	/	38	E e	60	[{
17	0	39	F f	61	\
18	1	40	G g	62	] }
19	2	41	H h	63	^ ~
20	3	42	I i	64	_
21	4	43	J j	65	
22	5	44	K k		

***NOTE:** DEC Multinational Characters are sorted with other characters. All a's are grouped with other a's, b's with b's, and so on. In foreign languages, the sort order is often different from the order used for English. Technical characters are sorted after standard characters in ascending order, and before standard characters in descending order. See WPS-PLUS Editor Functions for more information about multinational and technical characters.*

Sorting Missing Fields or Field Values

When you sort a list document by a field that is missing from some records, or by a field without a field value in some records, those records are:

- Placed first if the sort order is descending (reverse alphabetical order)
- Placed last if the sort order is ascending (alphabetical order)

Words and Phrases in a Sort Specification: Summary

Figure 6-7 shows the words and phrases used in a sort specification document.

```
sort specification (AND
sort specification,...)

where sort specification is one of the following:

SORT ON field name in ASCENDING ORDER
SORT ON field name in DESCENDING ORDER
```

Figure 6-7 Words and Phrases in a Sort Specification

Table 6-2 summarizes the words and phrases used in a sort specification document.

Table 6-2 Words and Phrases in a Sort Specification

Use	To
SORT ON field name IN ASCENDING ORDER	Sort list document records in ascending alphabetic or numeric order by the value in the specified record field.
SORT ON field name IN DESCENDING ORDER	Sort list document records in descending alphabetic or numeric order by the value in the specified record field.
AND sort specification	Use more than one sort specification. The later sort specification is used for sorting after the earlier one is used.

Running List and Sort Processing

This chapter explains how to run List and Sort Processing. In addition, it explains how to test list and sort specification documents without running List or Sort Processing.

You run List Processing or test a list specification document from the List Processing Menu.

To display the List Processing Menu:

Enter LP (List Processing) from the Document Processing Menu.

Figure 7-1 shows the List Processing Menu.

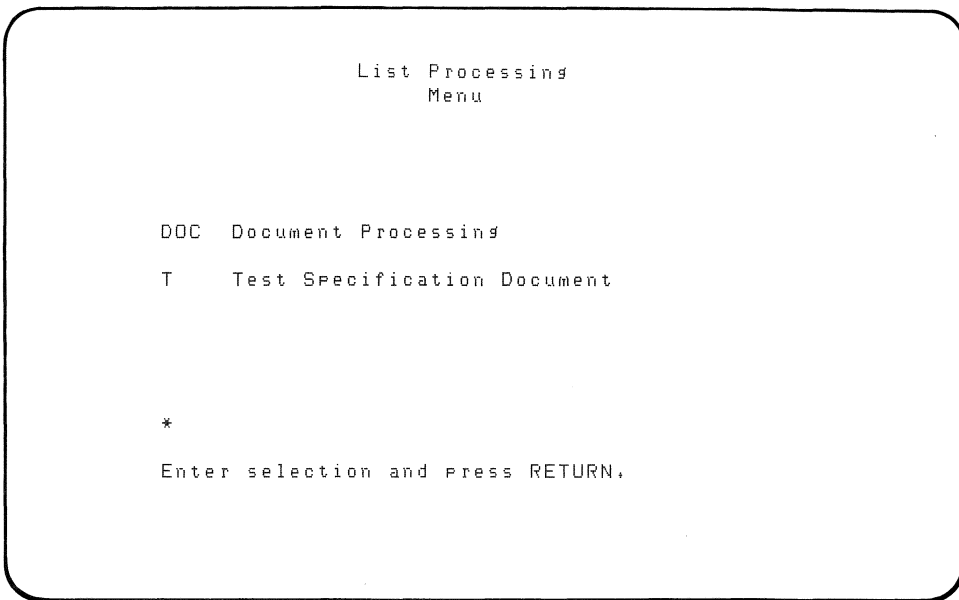


Figure 7-1 List Processing Menu

You run Sort Processing or test a sort specification document from the Sort Processing Menu.

To display the Sort Processing Menu:

Enter SR (Sort Processing) from the Document Processing Menu.

How to Run List Processing

To use List Processing:

- 1 Create the list, form, and list specification documents discussed in Chapters 2, 3, and 4.
- 2 Enter LP (List Processing) from the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).

NOTE: *Do not remove or replace the diskette until prompted to do so.*

The List Processing Menu displays.

- 3 Enter DOC (Document Processing) from the List Processing Menu.

The List Processing Input Documents form displays.

See Figure 7-2.

- 4 Type the names or numbers of the documents you want to use.

For example, TESTDOC identifies the TESTDOC document. The drive and directory are specified by the current Document Index, which you see when you use the I (Index) option from the Document Processing Menu. You can include a different drive or directory, or both. See the section Identifying Documents.

Press SHIFT/Down Arrow (or RETURN) to advance to the next field. Press SHIFT/Up Arrow to back up a field.

At the bottom of the form are settings that control the first and last records processed.

To process all records in your list document:

Leave the settings as they are.

To restrict the records processed:

Change the settings so they indicate the first and last records to process.

See the section Changing Start and Finish Settings.

- 5 Press RETURN after you have entered information in the last field on the screen, or press SHIFT/DO from any field.

The List Processing Output Document Menu displays.

- 6 Enter an option for your output document.

See Table 7-1.

- 7 Enter the name or number of the document you want to produce.

The List Processing Job Description form displays.

- 8 Make sure the documents displayed are correct.

To change documents:

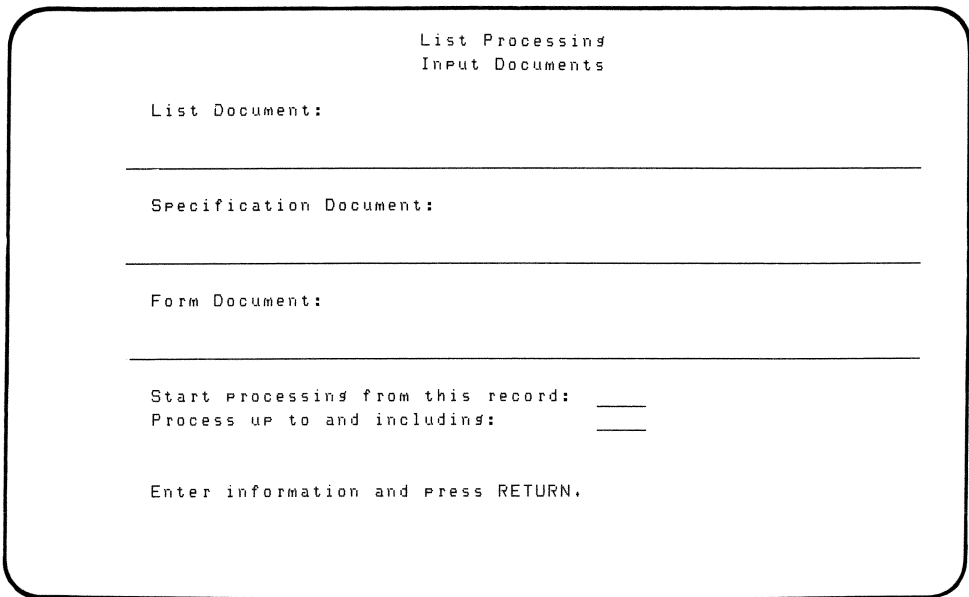
- Press EXIT SCREEN to display the List Processing Menu.
 - Enter DOC (Document Processing) and type the correct information on the List Processing Input Documents form.
- 9 Press RETURN at the List Processing Job Description form to start List Processing.

NOTE: You cannot use your terminal for other tasks while List Processing is running.

When List Processing is completed, WPS-PLUS displays the List Processing Menu. Press EXIT SCREEN to return to the Document Processing Menu.

Identifying Documents

The first display that appears after you select the DOC (Document Processing) option is the List Processing Input Documents form. Figure 7-2 shows the List Processing Input Documents form.



The screenshot shows a terminal window with the title "List Processing Input Documents". It contains several prompts for user input, each followed by a horizontal line for text entry. The prompts are: "List Document:", "Specification Document:", "Form Document:", "Start Processing from this record:", "Process up to and including:", and "Enter information and press RETURN.".

```
List Processing
Input Documents

List Document:
_____

Specification Document:
_____

Form Document:
_____

Start Processing from this record: ____
Process up to and including:      ____

Enter information and press RETURN.
```

Figure 7-2 List Processing Input Documents Form

You must identify three documents on this form:

- List document
- Form document
- List specification document

You can identify a document by:

- Document title
- Document number

Use the I (Index) option from the Document Processing Menu to obtain the document number from the Document Index.

You can include the drive or directory, or both, to identify a document. Include as much information as needed to identify the document you want.

If all of your WPS-PLUS documents are in the same directory, usually the document title is enough. For example:

```
TEST DOC
```

WPS-PLUS uses the drive and directory in your current Document Index to find this document. If a document is on a different drive, or in a different directory, include drive or directory when you specify the documents.

For example:

```
B:TESTDOC
```

This example identifies the TESTDOC document on drive B.

The next example identifies document 10 in the TEST directory:

```
\TEST\10
```

You can include both the drive and directory, if you need to:

```
E:\TEST\TESTDOC
```

This is a complete document path name. See *WPS-PLUS on PC* and the documentation for your PC for more information about path names.

Changing Start and Finish Settings

There are two settings on the List Processing Input Documents form:

```
Start Processing from this record:  ___
Process up to and including:       ___
```

The default values are:

```
Start Processing from this record: 1
Process up to and including:       0
```

The 0 setting has a special meaning in this context. The 0 setting ensures that WPS-PLUS processes every record in the list document that meets the condition(s) in the list specification document.

You can use the start and finish settings to restrict the records processed. For example, if 15 and 22 are your settings, only the 15th through the 22nd records of the list document are submitted to List Processing for possible selection. For List Processing to use a record, the record must be in the range from 15 through 22 and it must satisfy any conditions in the list specification document.

WPS-PLUS resets the two settings to their defaults when List Processing is finished.

When you complete the List Processing Input Documents form, the List Processing Output Document Menu displays.

Selecting an Option for Your Output

The options on the List Processing Output Document Menu control whether you create a new output document, or how you add text to an existing document. Table 7-1 describes these options:

Table 7-1 List Processing Output Document Menu Options

Use	To
O	Overwrite the current output document. The document you identify receives the output. Any text already in the document is destroyed. If no records are processed, you are left with an empty output document.
B	Add text to the bottom of the current output document. The document you identify receives the output. Any text already in the document remains, above the new output.
T	Add text to the top of the current output document. The document you identify receives the output. Any text already in the document remains, below the new output.
N	Create a new output document. The new document you identify receives the output.

If you enter option O, B, or T, you must enter the name of an existing output document in the space provided.

If you enter N, you must enter a name for a new output document. If you enter the name of a document that already exists, WPS-PLUS displays the message:

```
This document already exists
```

When you complete the menu, the List Processing Job Description form displays.

Validating Your Selections

Use the List Processing Job Description form to check that you selected the correct documents. Figure 7-3 shows the List Processing Job Description form.

The screenshot shows a text-based form titled "List Processing Job Description". It contains four sections for document information, each with fields for "Number", "Path", and "Title". The "List Document" section has Number 2, Path E:\WPSPLUS\, and Title SAMPLE LIST. The "Specification Document" section has Number 10, Path E:\WPSPLUS\, and Title SAMPLE LIST SPECIFICATION. The "Form Document" section has Number 12, Path E:\WPSPLUS\, and Title SAMPLE FORM. The "Output Document" section has Number 14, Path E:\WPSPLUS\, and Title SAMPLE OUTPUT. At the bottom, there are fields for "First record selected" (1) and "Last record selected" (0), followed by the instruction "Press RETURN to continue."

```

                                List Processing
                                Job Description

List Document:
  Number  2  Path E:\WPSPLUS\
  Title  SAMPLE LIST

Specification Document:
  Number 10  Path E:\WPSPLUS\
  Title  SAMPLE LIST SPECIFICATION

Form Document:
  Number 12  Path E:\WPSPLUS\
  Title  SAMPLE FORM

Output Document:
  Number 14  Path E:\WPSPLUS\
  Title  SAMPLE OUTPUT

First record selected: 1      Last record selected: 0
Press RETURN to continue.
```

Figure 7-3 List Processing Job Description Form

For each input and output document you select, the List Processing Job Description form displays:

- 1 The document number, assigned by WPS-PLUS when you create the document. The document number is unique in a directory.
- 2 The document title, assigned by you when you create a document.
- 3 The document path name. The document path name consists of the drive and directory (or directories) where the document is located. See *WPS-PLUS on PC* for more information about path names.

If a document in the display has the wrong name, probably more than one of your documents begins with the same sequence of letters.

To process the correct document:

- 1 Press EXIT SCREEN to return to the Document Processing Menu.
- 2 Enter I (Index) to obtain the document number from the Document Index.
- 3 Proceed through the List Processing displays again until you reach the List Processing Input Documents form.
- 4 Type the document number in place of the document title on the form.

If the document identified in the display is on the wrong drive, or in the wrong directory, include the correct drive or directory when you identify the document.

Press RETURN to start List Processing.

See the documentation for your PC for information about creating directories.

Information Displayed During List Processing

While you run List Processing, WPS-PLUS displays:

- The number of records selected
- The number of records processed
- The number of records in the list document

You can use this information to locate errors. For example, if you wanted to process all records in your list document, but the number of records selected is zero, there is probably an error in your list document.

To stop List Processing at any time:

Press ALT/INTERUPT.

How to Test a List Specification Document

To test your list specification document without running List Processing:

- 1 Enter LP (List Processing) from the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).
The List Processing Menu displays.
- 2 Enter T (Test Specification Document) at the List Processing Menu.
The Test Specification Document form displays.
- 3 Enter the name or number of the document you want to test. If necessary, include the drive and directory.
- 4 Press RETURN to start the test.

If there are errors in the specification document, an error message displays.

To correct the error:

Edit your list specification document.

How to Run Sort Processing

To use Sort Processing:

- 1 Create a sort specification document as discussed in Chapter 6.
You can use any list document you want to sort.
- 2 Enter SR (Sort Processing) from the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).
The Sort Processing Menu displays.
- 3 Enter SR (Sort Processing) from the Sort Processing Menu.
The Sort Processing Input Documents form displays.
- 4 Type the names or numbers of the documents you want to use. If necessary, include drives and directories.
See the section Identifying Documents.
Press SHIFT/Down Arrow (or RETURN) to advance to the next field.
Press SHIFT/Up Arrow to back up a field.
- 5 Press RETURN after you have entered information in the last field on the screen, or press SHIFT/DO from any field.
The Sort Processing Output Document Menu displays.
- 6 Enter an option for your output document.
See Table 7-1.
- 7 Enter the name or number of the document you want to produce.
The Sort Processing Job Description form displays.
- 8 Make sure the documents displayed are correct.
To change documents:
 - Press EXIT SCREEN to display the Sort Processing Menu.
 - Enter SR (Sort Processing) and type the correct information on the Sort Processing Input Documents form.
- 9 Press RETURN at the Sort Processing Job Description form to start Sort Processing.

NOTE: *You cannot use your terminal for other tasks while Sort Processing is running.*

When Sort Processing is completed, WPS-PLUS displays the Sort Processing Menu. Press EXIT SCREEN to return to the Document Processing Menu.

While you run Sort Processing, WPS-PLUS displays a running tally of:

- The number of records selected
- The number of records processed

In addition, WPS-PLUS displays:

- The number of fields used in sorting
- The number of records in the list document

You can use this information to locate errors. For example, if you want to sort by two fields, but WPS-PLUS shows 1 as the number of fields specified for sorting, the specification document may contain an error.

To stop Sort Processing at any time:

Press ALT/INTERRUPT.

How to Test a Sort Specification Document

To test your sort specification document without running Sort Processing:

- 1 Enter SR (Sort Processing) at the Document Processing Menu (if necessary, swap the WPS-PLUS diskette as instructed).
The Sort Processing Menu displays.
- 2 Enter T (Test Specification Document) at the Sort Processing Menu.
The Test Specification Document form displays.
- 3 Enter the name or number of the document you want to test. If necessary, include the drive and directory.
- 4 Press RETURN.

If there are errors in the specification document, an error message displays.

To correct the error:

 Edit your sort specification document.

How to Print an Output Document

To print an output document:

- 1 Return to the Document Processing Menu.
- 2 Enter P (if necessary, swap the WPS-PLUS diskette as instructed).
- 3 Follow the instructions on your screen.

For information about print settings, see *WPS-PLUS on PC*.

Index

A

- Abbreviation document, 2-7
- Addressing envelopes
 - form document for, 3-7
- ALT/INTERRUPT, 7-10, 7-12
- Angle brackets
 - for testing numbers, 4-10
 - in field names, 1-3, 2-3
 - in field values, 4-10, 4-17

B

- BUT NOT IF Condition, 4-5

C

- Characters
 - multinational, 6-15
 - technical, 6-15
- Columns, 3-17
- Command
 - See* name of each command
- Comments
 - inserting in list documents, 2-5
- Continuous-form
 - envelopes, 3-7

- Continuous-form (Cont.)

- labels, 3-9

- Conventions, x

- Creating lists

- See* List document

- Customized format, 5-20

D

- Dates

- testing against a range, 4-13

- Directories

- creating, 3-10

- Document

- See also* Form document, List

- document, List

- specification document,

- Output document, Sort

- specification document

- abbreviation, 2-7

- form, 3-1

- library, 2-7

- list, 1-8, 2-1

- list processing output, 1-5, 3-1

- list specification, 4-1

- sort output, 1-9, 6-2

- sort specification, 1-8, 6-1, 6-2

Document Processing Menu
List Processing (LP) option, 7-2
Sort Processing (SR) option, 7-2

E

<!E> (end repetition) command,
3-12, 3-20

Envelopes

as form documents, 3-3
continuous-form, 3-7
form document for, 3-7

Errors

List Processing, 7-10
List Processing Math, 5-23
Sort Processing, 7-12

F

Field name

changing, 3-20
definition, 1-3
guidelines, 2-3

Field value

definition, 1-3
guidelines, 2-3
missing, 2-5

Fields

accumulator, 5-14
definition, 1-3, 2-2
input, 5-8
missing, 2-5
numeric, 4-9, 5-8
required for list document, 2-4
result, 5-8

Finish setting, 7-6

Form document, 3-1

columns in, 3-17
creating, 3-2
definition, 1-4
leaders in, 3-12

Form document (Cont.)

summary of commands, 3-20
trailers in, 3-12
types of, 3-3

Form document commands

<> command, 3-15, 3-20
<!E> command, 3-20
<!R> command, 3-19, 3-20
<!S> command, 3-20

Form letters

formatting, 3-3

Formatting

See also Form document

envelopes, 3-7
form letters, 3-3
labels, 3-9
leaders, 3-12
mailing lists, 3-10
tables, 3-10, 3-14
with columns, 3-17
with double spacing, 3-16
with single spacing, 3-16
trailers, 3-12

FORMULA command, 5-7

checklist, 5-24
guidelines, 5-7, 5-8

FORMULA control block

creating, 5-5
sample, 5-3

G

<> (get-next-record) command,
3-15, 3-20

H

Headers and Footers

See Leaders, Trailers

L

Labels

- as form documents, 3-3
- form document, 3-9

Leaders, 3-12

- spacing, 3-14

Letterhead letters

- as form documents, 3-3

Letters

- as form documents, 3-3

Library document, 2-7

List document

- changing field names in, 3-20
- COMMENT command, 2-5
- comments in, 2-5
- creating, 2-2
- definition, 1-2
- dividing large ones, 2-6
- hints for handling, 2-6
- maintaining, 2-6
- making sublists, 3-19
- missing field values, 2-5
- missing fields, 2-5
- moving records in, 2-6
- necessary fields, 2-4
- order of fields, 2-4
- order of records, 2-6
- repeated areas, 3-12
- skipping records, 3-18

List Processing

- See also* List document, Form document, List specification document, Output document
- definition, 1-1
- documents required, 1-2
- errors, 7-10
- number of records in list document, 7-10
- number of records processed, 7-10

List Processing (Cont.)

- number of records selected, 7-10
- running, 7-1
- running tally, 7-10
- starting, 7-4

List Processing Input Documents

- form, 7-3, 7-5
- finish setting, 7-6
- start setting, 7-6

List Processing Job Description

- form, 7-8

List Processing Math, 1-7, 5-1

- \$, 5-21
- „, 5-21
- ., 5-20
- 9, 5-20
- *, for even columns, 5-22
- (), for negative numbers, 5-22
- accumulator field, 5-14
- averages, 5-11
- characters in formulas, 5-8
- commands, 5-7
- compared with Editor Math, 5-2
- control blocks, 5-2, 5-3, 5-4, 5-5, 5-6
- creating FORMULA control blocks, 5-5
- creating null control blocks, 5-6
- customized format, 5-20
- decimal places, 5-20
- default format, 5-19
- definition, 1-7
- errors, 5-23
- format checklist, 5-25
- format string, 5-19
- FORMULA command, 5-7
- FORMULA commands and control blocks, 5-3
- FORMULA control block, 5-4, 5-5
- input fields, 5-8

List Processing Math (Cont.)

- null control block, 5-6
- numeric fields, 5-8
- records selected for processing, 5-4
- result fields, 5-8
- ROUND command, 5-7, 5-19
- sample FORMULA control block, 5-2
- SET command, 5-7
- starting, 5-4
- stopping, 5-6
- subtotals, 5-15
- totals, 5-14
- TRUNCATE command, 5-7, 5-18
- using, 5-2
- WPSMATH command, 5-7

List Processing Math commands

- FORMULA command, 5-7
 - checklist, 5-24
 - guidelines, 5-7, 5-8
- ROUND command, 5-7
- SET command, 5-7
- TRUNCATE command, 5-7
- WPSMATH command, 5-7

List Processing Math format

characters

- \$, 5-21
- „, 5-21
- ., 5-20
- 9, 5-20
- *, for even columns, 5-22
- (), for negative numbers, 5-22
- +, leading, 5-22
- , trailing, 5-21

List Processing Math results

- customized format of, 5-20
- default format of, 5-19

List Processing Menu, 7-1

- Document Processing (DOC)
 - option, 7-3
- Test Specification Document (T)

List Processing Menu (Cont.)

- option, 4-14, 7-10

List Processing Output Document Menu, 7-7

- B option, 7-7
- N option, 7-7
- O option, 7-7
- T option, 7-7

List specification document

- AND after BUT NOT IF, 4-5
- AND condition, 4-4
- angle brackets (<>), 4-10
- BUT NOT IF condition, 4-5
- creating, 4-2
- definition, 1-5
- errors in, 4-14, 7-10
- guidelines, 4-13
- IF condition, 4-2
- missing field names, 4-7
- multiple OR statements, 4-5
- OR condition, 4-3
- OR LESS, 4-11
- OR MORE, 4-11
- PROCESS RECORD, 4-2
- processing all records, 4-2
- rejecting records by field value, 4-5
- restricting records processed, 4-2
- selecting a record by more than one value, 4-4
- successive BUT NOT IFs, 4-5
- testing
 - character-for-character, 4-10
 - dates, 4-12
 - digits only, 4-10
 - for a specific number of characters, 4-7
 - for any single character, 4-7
 - for empty fields, 4-6
 - for errors in, 4-14, 7-10
 - for partial values, 4-7

List specification document
testing (Cont.)

 numbers only, 4-10

 numeric ranges, 4-11

 numeric values, 4-9

 THROUGH, 4-11

 <*> wild card, 4-7

 <?> wild card, 4-7

Lists, creating

See List document

M

Mailing list

 creating, 3-10

Multinational characters, 2-7, 6-15

N

Null control block

 creating, 5-6

 definition, 5-6

Numbers

 testing character-for-character,
 4-10

 testing ranges, 4-11

Numeric fields

 definition, 4-9

Numeric ranges, 4-11

O

Output document

 definition, 1-5

 double spacing in, 3-16

 grouping output in, 3-14

 leaders in, 3-12

 print, 7-13

 single spacing in, 3-16

 trailers in, 3-12

P

Path names, 7-6, 7-9

Printing

 output document, 7-13

Process up to and including, 7-6

Processing lists

See List Processing

R

<!R> (copy record) command, 3-19,
3-20

Ranges

 numeric tests, 4-11

Records

 definition, 1-3

 maintaining, 2-6

 rejecting, 4-5

Repetitive area

 definition, 3-12

ROUND command, 5-7

S

<!S> (start repetition) command,
3-12, 3-20

Selecting records for List Processing

See also List specification
 document

 finish setting, 7-6

 start setting, 7-6

SET command, 5-7

Sort output document, 6-2

Sort Processing, 1-8, 6-1, 7-11

See also List document, Sort
 specification document, Sort
 output document

 definition, 1-8

 errors, 7-12

 fields used in sorting, 7-12

Sort Processing (Cont.)

- missing field values, 6-15
- missing fields, 6-15
- number of records in list
 - document, 7-12
- on more than one field, 6-5
- running, 7-11
- running tally, 7-12
- sort alphabet, 6-14
- sort in ascending order, 6-2
- sort in descending order, 6-2

Sort Processing Input Documents

- form, 7-11

Sort Processing Job Description

- form, 7-11

Sort Processing Menu

Sort Processing (SR)

- option, 6-4, 7-11

Test Specification Document (T)

- option, 7-12

Sort Processing Output Document

- Menu, 7-11

Sort specification document, 6-1, 6-2

- AND, 6-5
- creating, 6-4
- errors, 7-12
- IN ASCENDING ORDER, 6-4
- IN DESCENDING ORDER, 6-4
- SORT ON, 6-4
- testing for errors in, 7-12
- words and phrases used in, 6-15

Sorting lists

See Sort Processing

Start processing from this record, 7-6

Start setting, 7-6

Storing field names, 2-7

T

Tables

- creating, 3-10
- spacing entries in, 3-14, 3-16

Technical characters, 2-7, 6-15

Testing

See also List specification

- document character-for-character, 4-10

- dates, 4-12

- digits only, 4-10

- for a specific number of characters, 4-7

- for any single character, 4-7

- for empty fields, 4-6

- for partial values, 4-7

- list specification documents, 4-14, 7-10

- numeric ranges, 4-11

- numeric values, 4-9

- sort specification documents, 7-12

Trailers, 3-12

- spacing, 3-14

TRUNCATE command, 5-7

W

Wild cards, 4-7

WPSMATH command, 5-7

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and Puerto Rico
call **800-258-1710**

In Canada
call **800-267-6146**

In New Hampshire,
Alaska or Hawaii
call **603-884-6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

Reader's Comments

Your comments and suggestions are welcome. They will help us in our continuous effort to improve the quality and usefulness of our publications.

1. How would you rate this manual for:

	good	fair	poor
completeness of information	_____	_____	_____
accuracy of information	_____	_____	_____
ease of use (clarity, organization)	_____	_____	_____

2. Did you find errors in this manual? Please specify by page and paragraph, or mark corrections on pages and attach copies to this form.

	page	paragraph
a) Incorrect information:	_____	_____
b) Information left out:	_____	_____
c) Hard to understand:	_____	_____

3. What suggestions do you have for improving this manual?

4. What parts of the manual were especially good at helping you understand things?

5. Please check the boxes that apply to you.

☐ Technical	☐ Used other word processors
☐ Nontechnical	☐ Used computers
☐ Management	☐ Use the word processing unit yourself
☐ Nonmanagement	
☐ Other (explain)_____	

Name_____ Title_____

Company_____

Street_____ City_____

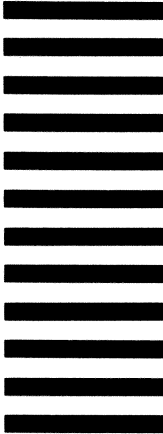
State_____ Zip_____ Date_____

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

WPS-PLUS Documentation Group
ZK02-2/R94
Office Systems Documentation
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NH 03062

-----Do Not Tear - Fold Here and Tape-----